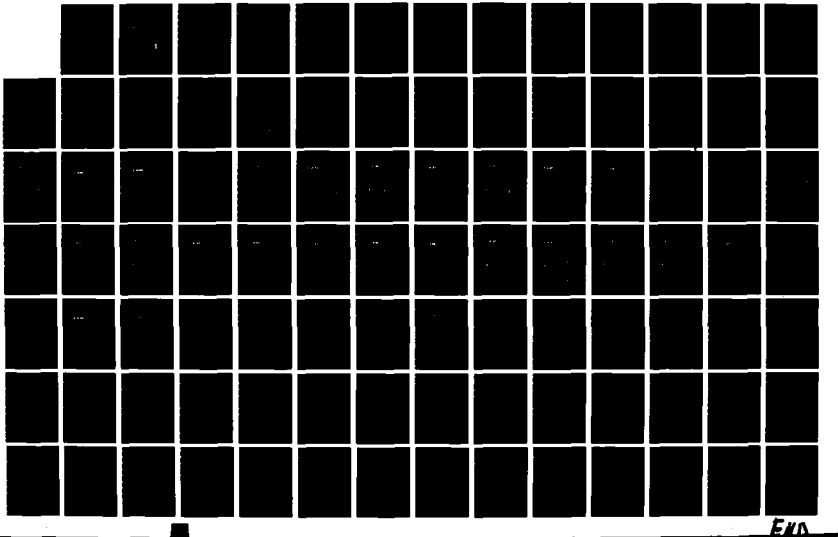
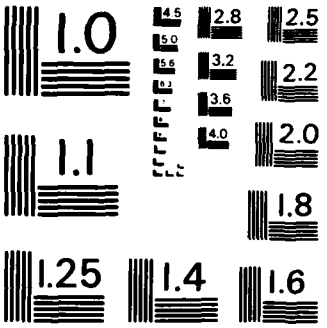
AD-A161 711 A STUDY OF GAUSS-SEIDEL-TYPE METHODS FOR SIMULATING LARGE-SCALE CIRCUITS(U) ILLINOIS UNIV AT URBANA COORDINATED SCIENCE LAB K C ANDERSON JAN 84 R-1014 1/1

UNCLASSIFIED N00014-79-C-0424 F/G 9/5 NL





REPORT R-1014 JANUARY 1984 UILU-ENG 84-2208

# A STUDY OF GAUSS-SEIDEL-TYPE METHODS FOR SIMULATING LARGE-SCALE CIRCUITS

KEITH CLARENCE ANDERSON

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AKL 711

## REPORT DOCUMENTATION PAGE

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS: None |
| 2a. SECURITY CLASSIFICATION AUTHORITY: N/A | 3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release, distribution unlimited. |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S): R-1014 UILU-ENG 84-2208 | 5. MONITORING ORGANIZATION REPORT NUMBER(S): N/A |
| 6a. NAME OF PERFORMING ORGANIZATION: Coordinated Science Laboratory, University of Illinois | 6b. OFFICE SYMBOL (If applicable): N/A |
| 7a. NAME OF MONITORING ORGANIZATION: Office of Naval Research | |
| 6c. ADDRESS (City, State and ZIP Code): University of Illinois at Urbana-Champaign, 1101 West Springfield Ave., Urbana, IL 61801 | 7b. ADDRESS (City, State and ZIP Code): 800 N. Quincy Street, Arlington, VA 22217 |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Joint Services Electronics Program | 8b. OFFICE SYMBOL (If applicable): N/A |
| 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: N00014-79-0424 | |
| 8c. ADDRESS (City, State and ZIP Code): 800 N. Quincy Street, Arlington, VA 22217 | 10. SOURCE OF FUNDING NOS. |

| PROGRAM ELEMENT NO. | PROJECT NO. | TASK NO. | WORK UNIT NO. |
|---|---|---|---|
| N/A | N/A | N/A | N/A |

11. TITLE (Include Security Classification): A STUDY OF GAUSS-SEIDEL-TYPE METHODS FOR SIMULATING LARGE-SCALE CIRCUITS

12. PERSONAL AUTHOR(S): Anderson, Keith Clarence

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Yr., Mo., Day) | 15. PAGE COUNT |
|---|---|---|---|
| Technical, final | FROM Jan '83 TO Dec '83 | January 1984 | 96 |

16. SUPPLEMENTARY NOTATION: N/A

| 17. COSATI CODES | | | 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUB. GR. | Integrated Circuits; time-domain simulation of MOS LSI circuits; Gauss-Seidel Method |
| | | | |
| | | | |

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Gauss-Seidel relaxation method is becoming widely used in the time-domain analysis of larege-scale integrated circuits. This research is concerned with the study of the stability properties of point Gauss-Seidel methods when applied to the simulation of MOS circuits. The stability properties of the standard Gauss-Seidel and of a modified Gauss-Seidel method with prediction are compared; and a new modified Gauss-Seidel method is proposed which incorporates the desirable properties of both the standard Gauss-Seidel and the one with prediction. The method has been tested in a computer program for the timing analysis of MOS circuits.

| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT | 21. ABSTRACT SECURITY CLASSIFICATION |
|---|---|
| UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐ | UNCLASSIFIED |

| 22a. NAME OF RESPONSIBLE INDIVIDUAL | 22b. TELEPHONE NUMBER (Include Area Code) | 22c. OFFICE SYMBOL |
|---|---|---|
| | | NONE |

DD FORM 1473 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

# A STUDY OF GAUSS-SEIDEL-TYPE METHODS FOR SIMULATING LARGE-SCALE CIRCUITS

BY

KEITH CLARENCE ANDERSON

B.S., University of Oklahoma, 1982

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1984



Urbana, Illinois

# CONTENTS

CHAPTER 4



CHAPTER 5



APPENDIX

# CHAPTER 1

## Introduction

The increase in the complexity of circuits fabricated on a single semiconductor chip has made the design and testing of the circuits more difficult. For a number of years, circuits with only a small number of transistors have been successfully analyzed using circuit simulators such as SPICE2[1]. However, as the size of the circuit grows, the execution time and memory requirements of these simulators become prohibitive. These problems have led to the development of new simulators that are much faster and less memory intensive than SPICE-type simulators such as MOTIS[2], MOTIS-C[3], and MOSTAP[4].

The new simulators developed for analyzing LSI and VLSI circuits often restrict the type of circuits that they can deal with. This knowledge is then utilized in a variety of ways to quickly produce relatively accurate time domain waveforms for the circuits. For digital circuits, this means that the process of design verification can be carried out on a level between that of conventional circuit analysis and conventional logic simulation.

One of the most common techniques used in large scale circuit simulators is the decomposition of the circuit into smaller subcircuits. Each subcircuit is then analyzed individually and the results are combined to give the desired data. In the general case, this is known as "tearing" and corresponds to partitioning the network equations in particular way[5]. If the subcircuits can be considered unidirectional (as is often the case for MOS circuits), then further gains in speed can be achieved by analysis sequencing and latency checking[6].

A second approach can be considered as temporal decomposition of the circuit. In this method, known as waveform relaxation, the solution for an entire time interval is approximated using only a limited number of iterations at any point within the interval. This

sweep of the whole interval can then be repeated until the solutions converge[7]. However, if it can be justified that the first sweep is accurate enough, it is possible to discard the following sweeps as well as the overhead associated with fully implementing waveform relaxation.

In timing simulators such as MOTIS-C, and SPLICE[8], a Gauss-Seidel-like technique is used to decouple the network equations and a single sweep of the relaxation technique is taken. This approach of not iterating to convergence means that the classical numerical properties of an algorithm such as stability and convergence may not hold. Hence, new studies of these properties become necessary to provide a basis for the use of an algorithm[9].

Wei proved in Reference [10] that the standard Gauss-Seidel method is not convergent when floating capacitors exist in the circuit. This helped motivate the development of a modified Gauss-Seidel method which used a forward predictor to estimate the values of unsolved variables in feedback loops. The predictor Gauss-Seidel method, shown to be zero stable and convergent, was implemented in a program called PREMOS designed for simulating NMOS circuits.

The work described in this thesis covers two basic areas. The first is a further study of the numerical properties of the standard Gauss-Seidel and the predictor Gauss-Seidel methods. A modification to the predictor method is made that gives it certain properties of both methods. The second area is concerned with adding the capability of analyzing some CMOS circuits to the PREMOS program.

Chapter 2 is a short description of some basic techniques used in numerical circuit analysis. The properties of the Gauss-Seidel, predictor Gauss-Seidel, and modified predictor Gauss-Seidel are studied in Chapter 3, while Chapter 4 describes the CMOS circuits that have been implemented in PREMOS. Finally Chapter 5 presents some conclusions and comments.

# CHAPTER 2

## Circuit Simulation Techniques

### 2.1 Introduction

Conventional circuit simulators have proven to be very successful at providing accurate current and voltage waveforms for small scale integrated circuits. However, in LSI and VLSI design, these exact waveforms are often not needed as much as just the timing of the transitions from one logic state to another. This led to the development of logic simulators[8] that give results in terms of discrete levels. These logic simulators are much faster than circuit simulators, but the loss of information is not always acceptable. In an attempt to bridge this gap between the accuracy of circuit simulators and the speed of logic simulators, timing simulators are being developed with the goal of providing waveforms close to those of circuit simulators at speeds approaching those of logic simulators.

Most timing simulators can be broken into two categories: those based on conventional circuit analysis that employ various techniques to increase the speed, and those based on logic simulation but employ various techniques to increase the accuracy. This report is concerned with techniques more closely related to circuit analysis than to logic simulation. In this respect, some of the existing circuit analysis techniques are reviewed in this chapter.

### 2.2 Conventional Circuit Analysis

A nonlinear dynamic circuit may in general be characterized by the equation,

$$f(x,\dot{x},t) = 0, \quad x(0) = x_0 \tag{2.1}$$

where x represents a vector of voltages and/or currents and t represents time. In order to obtain a numerical solution to x, Equation (2.1) is discretized at each time point $t_n$ by using

an integration formula such as the backward Euler, trapezoidal, or one of Gear's formulas. This discretization transforms Equation (2.1) at each time point to an algebraic equation of the form,

$$g(x_n) = 0 \tag{2.2}$$

Equation (2.2) is then usually solved by a modified Newton's method which repetitively develops and solves linear equations of the form,

$$A\,x = b \tag{2.3}$$

where A is a matrix and b is a vector. At every iteration A and b must be constructed by linearizing the nonlinear equation at a new iteration point found during the previous iteration. This process is repeated until the sequence of x converges to within some specified tolerance. Once the solution for $x_n$ at a this point is found, the time is incremented and the process is started again until the final time is reached. The basic algorithm can be given as

```
BEGIN
 BEGIN
  X = [Voltages, Currents]
  TIME = Start Time
  H = Initial Time step
 END {initialization}
 TIME = TIME + H
 WHILE (TIME < End Time) DO
  BEGIN
   Discretize the differential operators by
   using an integration formula.
  REPEAT
   BEGIN k = 1
      Evaluate linear models for circuit elements at
      the operating points and form the circuit
      matrix A and vector b.
      Solve linear equations AX = b.
   END
```

```
    UNTIL (convergence achieved) {dc loop}
    IF the local truncation error (LTE) is
      smaller than the tolerance
     THEN
        BEGIN
          Compute new time step H
          TIME = TIME + H
        END
     ELSE
        BEGIN
          TIME = TIME - H
          Compute revised time step H
          TIME = TIME + H
        END
   END {time loop}
END
```

An increase in speed can be achieved by using sparse matrix techniques. However, these techniques are not cost effective for VLSI simulations; as a result, new approaches have recently been proposed. The basis for several of these new methods will be described in the next sections.

### 2.3 Large Scale Circuit Analysis

A number of techniques based on relaxation methods have been proposed for solving large systems of simultaneous equations. When these techniques are applied to circuits, a priori knowledge of some of the properties of the circuits can be used to significantly decrease the amount of computation required while at the same time maintaining accuracy. However, it is important to first understand the basic methods.

### 2.4 Point Gauss-Jacobi Algorithm

If the vector x in Equation (2.2) satisfies $x \in R^m$ and $x_k^n$ is the value of the kth component of the vector x at time $t_n$, the value of $x_k^{n+1}$ is found by solving the scalar equation,

$$g(x_1^n, x_2^n, \ldots x_{k-1}^n, x_k, x_{k+1}^n, \ldots, x_m^n) \tag{2.4}$$

The complete vector $x^{n+1}$ is found by incrementing k from 1 to m.

While this algorithm is fairly easy to implement, it may converge to the solution very slowly, provided that it converges at all. A natural extension of this algorithm leads to the next algorithm.

### 2.5 Point Gauss-Seidel Algorithm

In this algorithm, information from the present iteration is used as well as information from the previous iteration which usually decreases the number of iterations necessary for convergence. When computing component $x_k^{n+1}$, the equation to be solved is

$$g(x_1^{n+1}, x_2^{n+1}, \ldots, x_{k-1}^{n+1}, x_k, x_{k+1}^{n}, \ldots x_m^{n}) \tag{2.5}$$

This algorithm can result in a considerable increase in speed, especially when the circuit variables are evaluated in the same order as the signal flow through the circuit.

### 2.6 Block Algorithms

If each $x_k$ represents a node voltage, then from the network point of view, the two methods above are equivalent to decomposing the network at every node. Another possibility is to decompose the circuit into subcircuits composed of several nodes. When the Gauss-Jacobi or Gauss-Seidel methods are applied in conjunction with decomposition into subcircuits, Equations (2.4) and (2.5) can still be used, but with each $x_k$ representing a vector instead of a scalar. In this case each vector $x_k$ can be solved using the conventional circuit analysis previously described.

### 2.7 Waveform Relaxation

For all of the algorithms described so far, the analysis is carried out at each time point for the entire circuit before proceeding to the next time point. However, it is also possible to solve for the waveform of each subcircuit over the entire time interval before proceeding to the next subcircuit. After finding the waveforms for all of the subcircuits, the process can be repeated until the waveforms converge to a solution[7].

One of the advantages of waveform relaxation is that different step sizes can be used for different subcircuits. However, a large amount of memory may be required to store the waveforms and a large number of iterations may be needed for the solution to converge.

# CHAPTER 3

## Numerical Properties

### 3.1 Introduction

The techniques of the previous chapter can all require a considerable amount of time to check to see if the computed solution is accurate enough. At each time point the algorithm will have to check for convergence and then decide whether another iteration is needed. Then after convergence for that time point is achieved, the local truncation error (LTE) must be computed so that a decision can be made as to the accuracy of the point in question. If the LTE is too large, the point may be thrown out and a new, smaller step must be taken before repeating the process. Even if the point is accepted, the next step size is usually recomputed before proceeding. The end result is that a lot of overhead computation not directly related to the desired solution is performed.

This has led to the development of some simulators that used a fixed number of iterations at each time point as well as a fixed step size. In fact some of these simulators such as MOTIS, use only a single iteration. While this approach can obviously save a lot of computation time on a per sweep basis, the fact that MOTIS only takes one sweep means that the relaxation is not carried to convergence and classical numerical properties such as stability and accuracy may not hold. Therefore, these properties must be evaluated to provide a proper basis for the use of this method.

### 3.2 Stability

In general, the analysis of a particular method is studied on a test problem simple enough to be analyzed theoretically, yet complex enough to provide information about a wide variety

of applications. The most common test problem used is probably the single time invariant linear differential equation:

$$\dot{x} = ax, \qquad x(0) = x_0 \tag{3.1}$$

where "x" and "a" are scalars. Using a single equation has worked well when an integration method like the backward Euler, trapezoidal, or one of Gear's formulas has been carried to convergence at each step. However when a Gauss-Seidel type technique is used and the process is not iterated to convergence at each step, the values of the variables behave differently according to the order in which they are processed. Consequently, the test problem must be generalized to become a system of equations,

$$C\dot{x} = G\,x, \qquad x(0) = x_0 \tag{3.2}$$

where x is a vector, C and G are matrices, and C is invertible. When an integration algorithm is applied to Equation (3.2) it is possible to express the value of x at $t_{n+1}$ recursively in the form,

$$x_{n+1} = M(h)\,x_n \tag{3.3}$$

where h is the step size and M(h) is known as the companion matrix. In terms of the initial conditions, Equation (3.3) becomes

$$x_{n+1} = [M(h)]^{n+1}\,x_0 \tag{3.4}$$

The numerical properties of one step integration algorithms are then defined in terms of the properties of the companion matrix.

Definition 3.1[10]

An integration algorithm is consistent if when applied to Equation (3.2), its companion matrix can be expanded in a power series as a function of the step size h as

$$M(h) = I + hA + 0(h^2) \tag{3.5}$$

where A equal $C^{-1}G$.

Theorem 3.1

For a given step size $h^*$, the sequence of vectors $\{x_n\}$ in Equation (3.4) is bounded if and only if the spectrum (or set of eigenvalues) of $M(h^*)$ is contained in the unit ball B(0,1) and no multiple zero of the minimal polynomial has modulus equal to one[11]. (The requirement that no multiple zero of the minimal polynomial has modulus equal to one is the same as saying that all the eigenvalues with a magnitude of one must be distinct.)

The above theorem leads to the following definition of stability for an integration algorithm:

Definition 3.2

An integration algorithm is zero stable if and only if there exists a $\delta > 0$ such that for all h in the interval $(0, \delta)$, the spectrum of M(h) is contained in the unit ball B(0,1), and no multiple zero of the minimal polynomial has modulus equal to one.

Note that if a circuit being evaluated is stable, then the solution is bounded, and the integration algorithm must also be stable if an accurate solution is desired.

Definition 3.3

Let x(t) be the exact solution of the test problem. An integration algorithm is convergent if the sequence of the computed solution converges uniformly to x(t) as the step size tends to zero.

Theorem 3.2

If an integration algorithm is consistent and zero stable, then it is convergent.

The proof of Theorem 3.2 can be found in numerical books such as Reference [12].

It has been proven in Reference [10] that the Gauss-Seidel method is convergent if there are no floating capacitors (i.e., capacitors connected between two nonground nodes) present in the circuit. However, it has also been proven in Reference [10] that the Gauss-Seidel method is not consistent, and therefore may not be convergent, whenever floating capacitors are present. This led Wei to introduce a modified Gauss-Seidel method which is convergent even when floating capacitors are present in the circuit. However, from the definition, convergence is dependent upon the step size tending to zero. In practice it is desirable to make the step size as large as possible for the desired degree of accuracy. Therefore, the range of h for which a method is stable is important, which is a different issue than convergence, and thus has to be studied separately. Another issue of importance is that the presence of complex conjugates in the spectrum of M(h) may indicate an oscillatory component in the computed solution that may not be present in the exact solution. These issues will be examined now by using a simple test circuit shown in Figure 3.1 to generate a system of equations and the corresponding companion matrices.

### 3.3 The Companion Matrices

Applying nodal analysis to the circuit of Figure 3.1, we obtain the matrix equation:

$$\begin{bmatrix} c_1+c_3 & -c_3 \\ -c_3 & c_2+c_3 \end{bmatrix} \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} + \begin{bmatrix} g_1+g_3 & g_m-g_3 \\ g_m-g_3 & g_2+g_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0 \qquad (3.6)$$

Letting

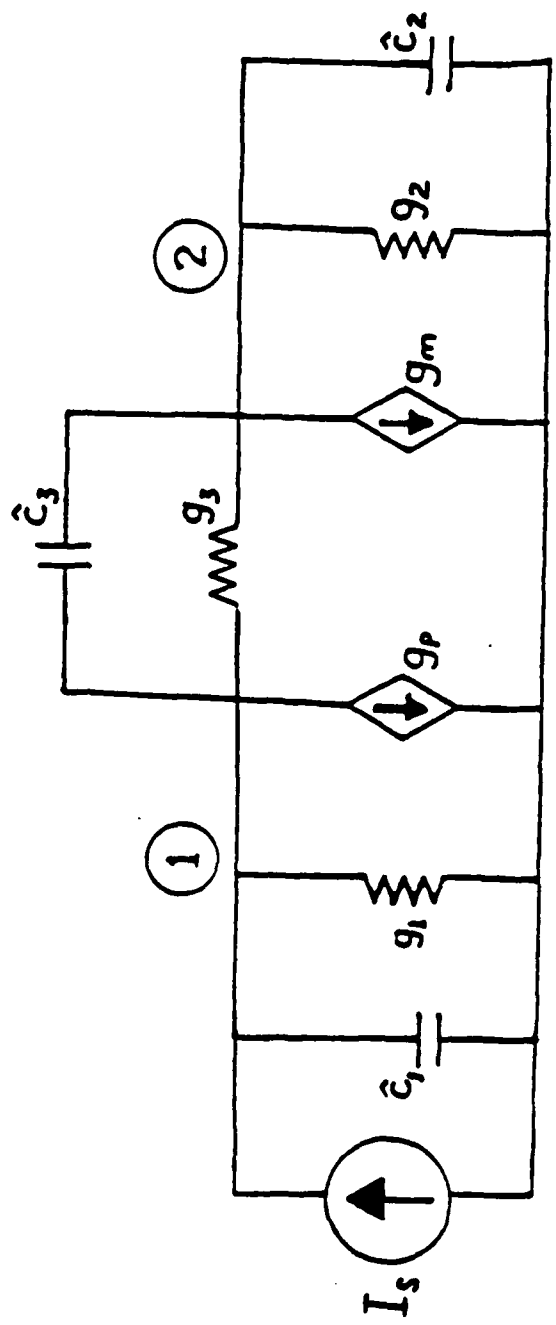



Figure 3.1 Test Circuit.

$$c_1 = \ell_1 + \ell_3 \tag{3.7}$$

$$c_2 = \ell_2 + \ell_3 \tag{3.8}$$

$$c_3 = \ell_3 \tag{3.9}$$

$$g_{11} = g_1 + g_3 \tag{3.10}$$

$$g_{12} = g_p - g_3 \tag{3.11}$$

$$g_{21} = g_m - g_3 \tag{3.12}$$

$$g_{22} = g_2 + g_3 \tag{3.13}$$

and

$$C = \begin{bmatrix} c_1 & -c_3 \\ -c_3 & c_2 \end{bmatrix} \tag{3.14}$$

$$G = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \tag{3.15}$$

$$V = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{3.16}$$

Equation (3.6) can be written in the form of Equation (3.2) as:

$$\dot{V} = -C^{-1} G V \tag{3.17a}$$

$$= A V \tag{3.17b}$$

If the backward Euler integration formula

$$\dot{x}_n = (x_n - x_{n-1})/h \tag{3.18}$$

is applied to Equation (3.6) we have

$$\begin{bmatrix} c_1 & -c_3 \\ -c_3 & c_2 \end{bmatrix} \begin{bmatrix} \dfrac{v_{1,n} - v_{1,n-1}}{h} \\ \dfrac{v_{2,n} - v_{2,n-1}}{h} \end{bmatrix} + \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} v_{1,n} \\ v_{2,n} \end{bmatrix} = 0 \tag{3.19a}$$

$$\begin{bmatrix} c_1 & -c_3 \\ -c_3 & c_2 \end{bmatrix} \begin{bmatrix} v_{1,n} \\ v_{2,n} \end{bmatrix} + \begin{bmatrix} hg_{11} & hg_{12} \\ hg_{21} & hg_{22} \end{bmatrix} \begin{bmatrix} v_{1,n} \\ v_{2,n} \end{bmatrix} - \begin{bmatrix} c_1 & -c_3 \\ -c_3 & c_2 \end{bmatrix} \begin{bmatrix} v_{1,n-1} \\ v_{2,n-1} \end{bmatrix} = 0 \quad (3.19b)$$

$$\begin{bmatrix} c_1+hg_{11} & -c_3+hg_{12} \\ -c_3+hg_{21} & c_2+hg_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_n - \begin{bmatrix} c_1 & -c_3 \\ -c_3 & c_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{n-1} = 0 \quad (3.19c)$$

At this point the derivation for the companion matrices of a full matrix solution, the standard Gauss-Seidel and the predictor Gauss-Seidel methods differ. For a full matrix solution this becomes

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_n = \begin{bmatrix} c_1+hg_{11} & -c_3+hg_{12} \\ -c_3+hg_{21} & c_2+hg_{22} \end{bmatrix}^{-1} \begin{bmatrix} c_1 & -c_3 \\ -c_3 & c_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{n-1} \quad (3.20a)$$

$$= \frac{1}{\Delta_k} \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{n-1} \quad (3.20b)$$

where

$$k_{11} = c_1 (c_2+hg_{22}) - c_3(c_3-hg_{12}) \quad (3.21)$$

$$k_{12} = -c_3(c_2+hg_{22}) + c_2(c_3-hg_{12}) \quad (3.22)$$

$$k_{21} = c_1(c_3-hg_{21}) - c_3(c_1+hg_{11}) \quad (3.23)$$

$$k_{22} = -c_3(c_3-hg_{21}) + c_2(c_1+hg_{11}) \quad (3.24)$$

$$\Delta_k = (c_1+hg_{11})(c_2+hg_{22}) - (-c_3+hg_{12})(-c_3+hg_{21}) \quad (3.25)$$

### *3.3.1 Standard Gauss-Seidel*

When applying the standard Gauss-Seidel technique, Equation (3.19) becomes

$$\begin{bmatrix} c_1+hg_{11} & 0 \\ -c_3+hg_{21} & c_2+hg_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_n = \begin{bmatrix} c_1 & -c_3 \\ -c_3 & c_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{n-1} - \begin{bmatrix} 0 & -c_3+hg_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{n-1} \quad (3.26a)$$

$$= \begin{bmatrix} c_1 - hg_{12} \\ -c_3 \; c_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{n-1} \quad (3.26b)$$

Solving for $V_{1,n}$ and $V_{2,n}$,

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_n = \frac{1}{\Delta_m} \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{n-1} \tag{3.27}$$

where

$$m_{11} = c_1(c_2+hg_{22}) \tag{3.28}$$

$$m_{12} = -hg_{12}(c_2+hg_{22}) \tag{3.29}$$

$$m_{21} = c_1(c_3-hg_{21})-c_3(c_1+hg_{22}) \tag{3.30}$$

$$m_{22} = -c_3(c_3-hg_{21}) + c_2(c_1+hg_{11}) \tag{3.31}$$

$$\Delta_m = -c_3(c_3-hg_{21}) + c_2(c_1+hg_{11}) \tag{3.32}$$

The spectrum of $M_{GS}(h)$ can then be found as the roots of the equation,

$$\lambda^2 - ((m_{11} + m_{22})/\Delta_m)\lambda + ((m_{12}m_{21})/\Delta_m^2) = 0 \tag{3.33}$$

### *3.3.2 Predictor Gauss-Seidel*

In Reference [10] Wei introduced a modified Gauss-Seidel method which used a forward predictor for unsolved node voltages. The voltages were predicted according to the formula,

$$V^n_{N(guess)} = V^{n-1}_N + h_n \left( \frac{V^{n-1}_N - V^{n-2}_n}{h_{n-1}} \right) \tag{3.34}$$

For a more general formula which generates a family of equations and includes the original predictor method as a special case, the following predictor equation is proposed

$$V^n_{N(guess)} = V^{n-1}_N + \frac{h_n}{1+\gamma h_n} \left( \frac{V^{n-1}_N - V^{n-2}_N}{h_{n-1}} \right) \tag{3.35}$$

This family of equations obviously reduces to Equation (3.34) when gamma equals zero, but approaches the standard Gauss-Seidel method as gamma goes to infinity. Letting

$$\theta = \frac{1}{1+\gamma h} \tag{3.36}$$

applying Equation (3.35) to Equation (3.19), and assuming that $h_n = h_{n-1} = h$, we obtain

$$\begin{bmatrix} c_1+hg_{11} & 0 \\ -c_3+hg_{21} & c_2+hg_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_n = \begin{bmatrix} c_1 & -hg_{12}+\theta(c_3-hg_{12}) \\ -c_3 & c_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{n-1} \tag{3.37}$$

$$+ \begin{bmatrix} 0 & -\theta(c_3-hg_{12}) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{n-2}$$

or

$$\begin{bmatrix} c_1+hg_{11} & 0 & 0 \\ -c_3+hg_{21} & c_2+hg_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{1,n} \\ v_{2,n} \\ v_{2,n-1} \end{bmatrix} = \begin{bmatrix} c_1 & -hg_{12}+\theta(c_3-hg_{12}) & -\theta(c_3-hg_{12}) \\ -c_3 & c_2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_{1,n-1} \\ v_{2,n-1} \\ v_{2,n-2} \end{bmatrix} \tag{3.38}$$

The predictor companion matrix, $M_{PCS}(h)$ is found as

$$\begin{bmatrix} c_1+hg_{11} & 0 & 0 \\ -c_3+hg_{21} & c_2+hg_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} c_1 & -hg_{12}+\theta(c_3-hg_{12}) & -\theta(c_3-hg_{12}) \\ -c_3 & c_2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.39}$$

$$= \begin{bmatrix} P & Q & R \\ S & T & U \\ 0 & 1 & 0 \end{bmatrix}$$

where

$$P = c_1/\beta_1 \tag{3.40}$$

$$Q = (-hg_{12} + \theta(c_3-hg_{12}))/\beta_1 \tag{3.41}$$

$$R = (-\theta(c_3-hg_{12}))/\beta_1 \tag{3.42}$$

$$S = c_1(c_3-hg_{21})/(\beta_1\beta_2) - c_2/\beta_2 \tag{3.43}$$

$$T = (c_3-hg_{21})(-hg_{12} + \theta(c_3-hg_{12}))/(\beta_1\beta_2) + c_2/\beta_2 \tag{3.44}$$

$$U = (-\theta(c_3-hg_{21})(c_3-hg_{12}))/(\beta_1\beta_2) \tag{3.45}$$

$$\beta_1 = (c_1+hg_{11}) \tag{3.46}$$

$$\beta_2 = (c_2+hg_{22}) \tag{3.47}$$

The spectrum of $M_{PGS}(h)$ is then found as the roots to the cubic equation,

$$\lambda^3 - (P+T)\lambda^2 + (PT-SQ-U)\lambda + (PU-SR) = 0 \tag{3.48}$$

Given a cubic equation, an explicit solution in terms of the coefficients is found as follows[13]. A cubic equation of the form

$$y^3 + py^2 + qy + r = 0$$

may be reduced to the form

$$x^3 + ax + b = 0$$

by substituting (x - p/3) for y. When this is done, then

$$a = (3q - p^2)/3$$

and

$$b = (2p^3 - 9pq + 27r)/27$$

For solution let

$$A = \sqrt[3]{(-b/2) + \sqrt{(b^2/4 + a^3/27)}}$$

and

$$B = \sqrt[3]{(-b/2) - \sqrt{(b^2/4 + a^3/37)}}$$

Finally, the values of x will be given by

$$x = A+B,\ -(A+B)/2 + ((A-B)/2)\sqrt{-3},\ -(A+B)/2 - ((A-B)/2)\sqrt{-3}$$

When the above formulas are applied to Equation (3.48) the expressions obtained are complicated enough that the various numerical properties of the method are probably masked. Instead, the numerical properties of the methods when applied to the test circuit of Figure 3.1 with different parameter values are investigated.

### 3.4 Stability and Accuracy Studies

In this section several studies of the circuit given in Figure 3.1 will be presented. The component values in each case are selected so that the circuit is stable with strictly real eigenvalues and then the spectrum of each companion matrix as a function of the step size is plotted. Each plot will have a solid ellipse representing the upper half of the unit circle and a series of discrete points that give the spectrum at specific values of h. The values of the components will not be realistic for practical circuits, but by proper scaling these results should be applicable to a wide variety of cases. It can be seen from Equation (3.19) that if each capacitance and the step size h are multiplied by the same factor $\alpha_1$, then the results should remain unchanged. Similarly, if each conductance is multiplied by a factor $\alpha_2$, and the

step size is divided by $\alpha_2$ then once again the results should be the same. In summary, if the system

$$C\dot{x} + Gx = 0 \tag{3.49}$$

results in the companion matrix M(h), then the system

$$\alpha_1 C\dot{x} + \alpha_2 Gx = 0 \tag{3.50}$$

will give the companion matrix $M(\frac{\alpha_1}{\alpha_2} h)$.

The accuracy studies will be conducted by exciting node 1 of the circuit with a current source of 1.0 amp for all time $t > 0.0$ (with 0.0 amps as the input prior to $t = 0.0$). This step input will allow the response of the methods to a rapidly changing input to be compared by observing the voltage computed for node 2.

*3.4.1 Study A*

By letting $c_1 = c_2 = c_3 = 1$ *(farad) and* $g_1 = g_2 = g_3 = g_m = g_p = 1$ *(mho)* this study provides a reference to which other studies can easily be compared. For this circuit with eigenvalues -2/3 and -2, Figure 3.2 shows that both the standard and predictor Gauss-Seidel methods are stable for $0 < h < \infty$. The standard method, shown in Figure 3.2a, is strictly real for all h. The spectrum of the pure predictor method, (i.e., gamma = 0.0 in Equation 3.36) shown in Figure 3.2b has complex components for h greater than one-third and reaching a maximum at about h = 1. Finally, the modified predictor spectrum is shown in Figure 3.2c with gamma = 1.0. As expected, this spectrum appears to be a weighted average of the previous two.

Figures 3.3 and 3.4 are plots of the voltage at node 2 for h = 1.0 and h = 0.1 respectively. Comparing the standard and predictor Gauss-Seidel (gamma = 0.0) methods with the solution generated by a full matrix approach, we see in this case that all three

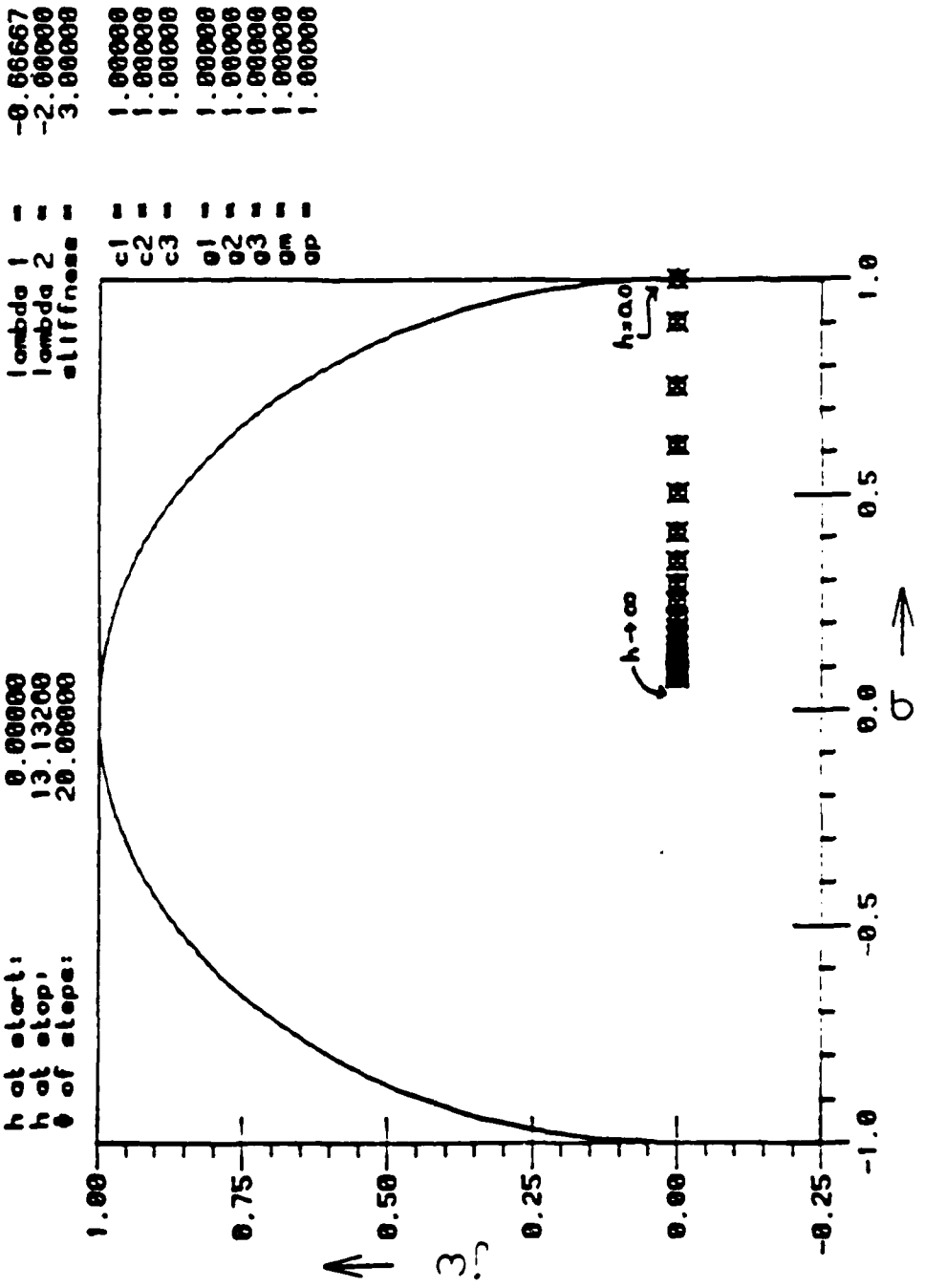



**Figure 3.2a** Standard Gauss-Seidel Spectrum for Study A.

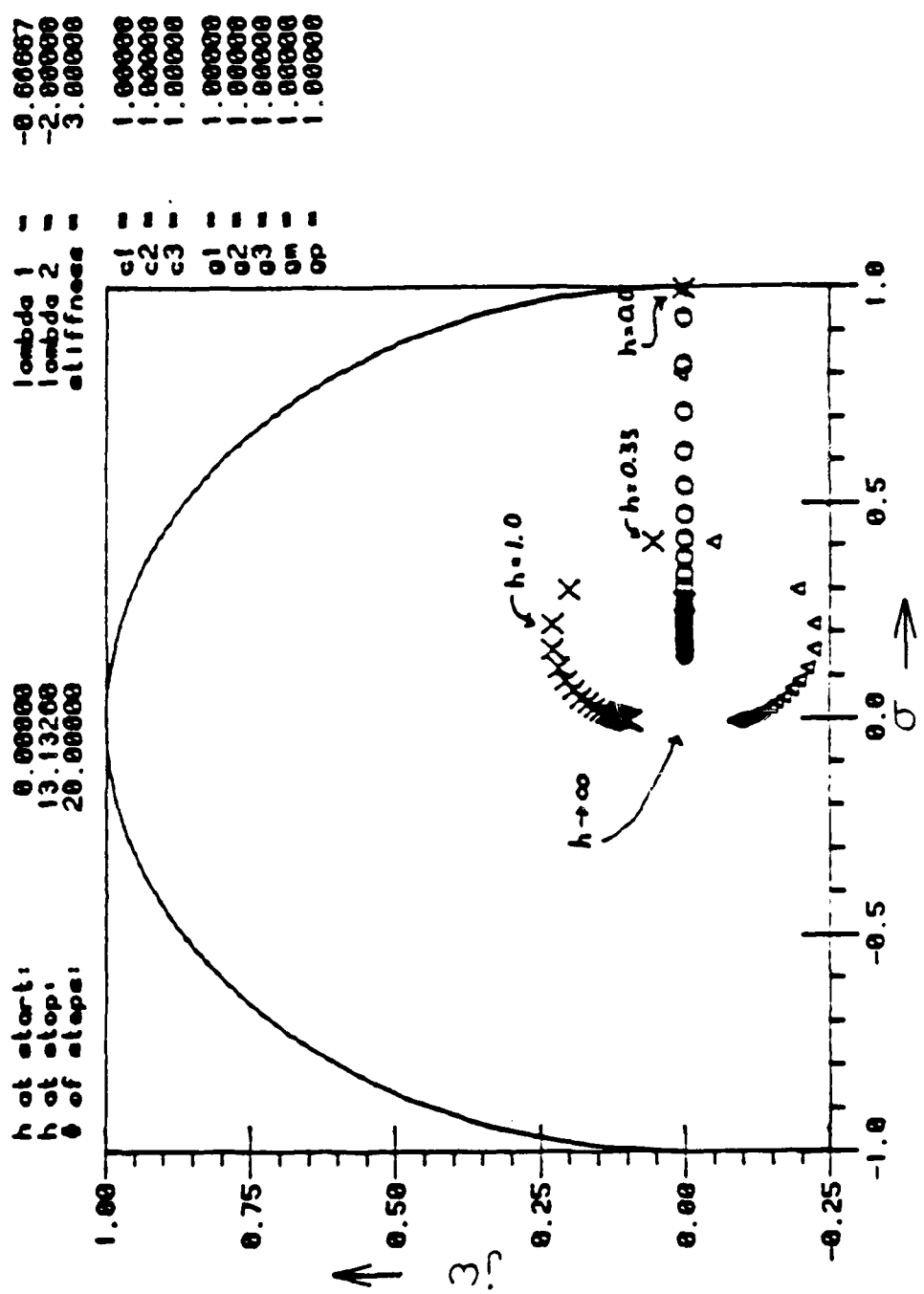



Figure 3.2b Predictor Gauss-Seidel Spectrum with gamma = 0.0 for Study A.

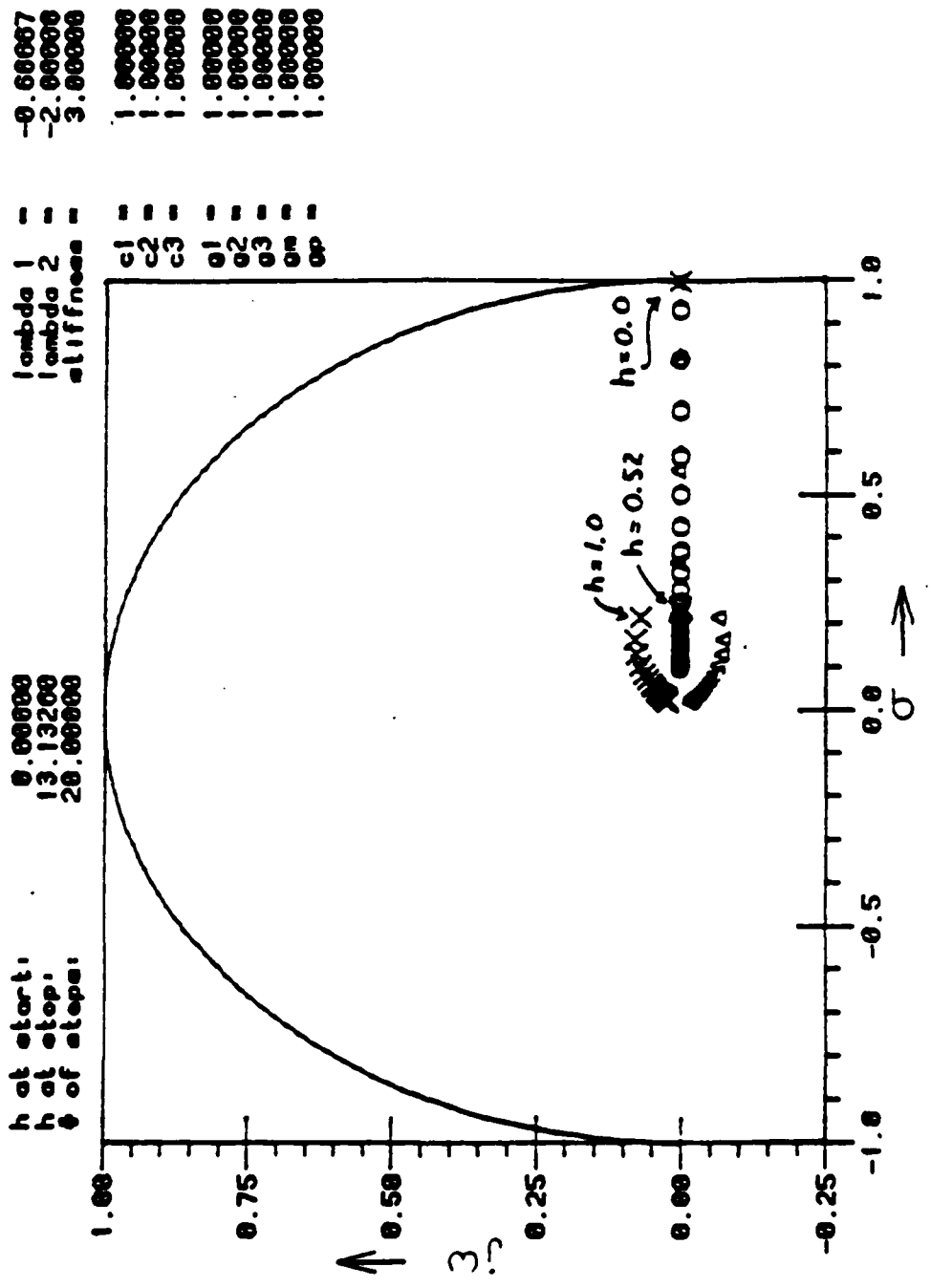



Figure 3.2c Predictor Gauss-Seidel Spectrum with gamma = 1.0 for Study A.

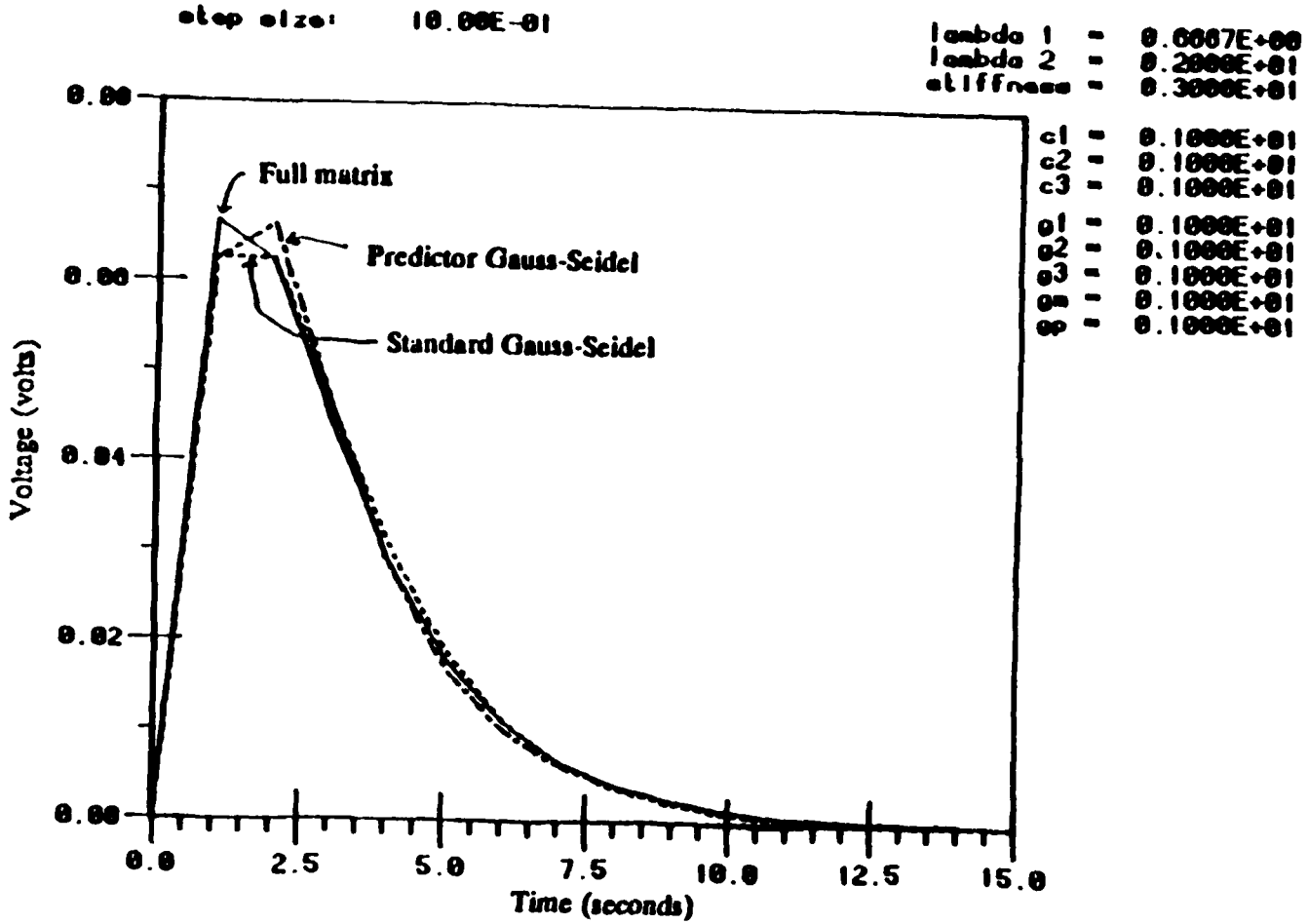



Figure 3.3 Node 2 output voltage with step size h = 1.0 and gamma = 0.0 for Study A.

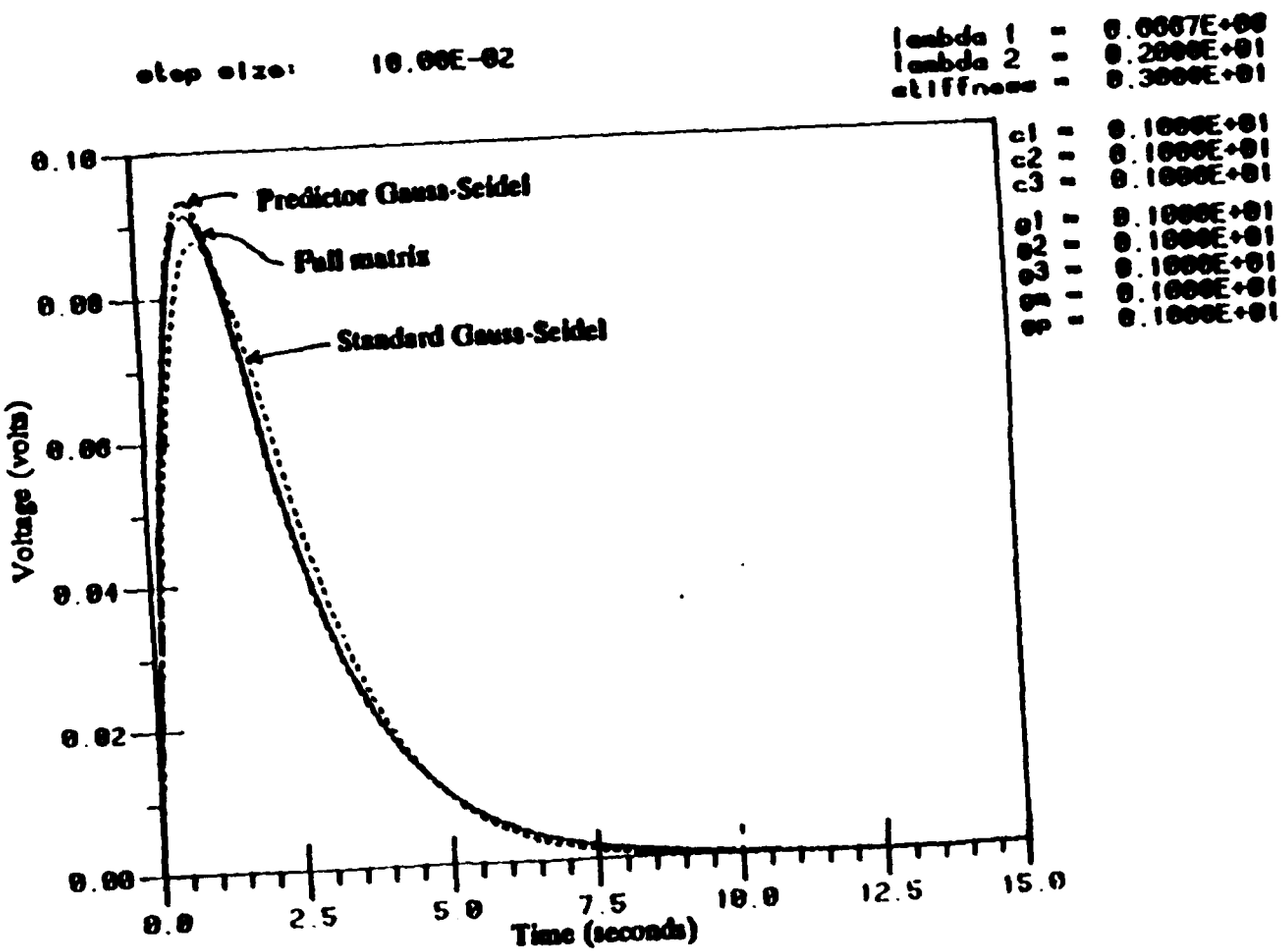



Figure 3.4 Node 2 output voltage with step size h = 0.1 and gamma = 0.0 for Study A.

methods have comparable accuracy for a given step size. Comparing the solutions generated by the two different step sizes, the larger step size does not allow the voltage to change as quickly as the smaller step size. Sharp corners and the lagging response in Figure 3.3 indicate that a step size (h = 1.0) of the same order of magnitude as the eigenvalues (-2/3 and -2) may result in an inaccurate solution when there is a rapidly changing input.

*3.4.2 Study B*

In this example, $C_3$ is reduced by a factor of ten. This has the effect of decreasing the stiffness of the circuit (i.e., ratio of the eigenvalues) and allows the spectrum of the standard method, Figure 3.5a and that of the predictor method (gamma = 0.0), Figure 3.5b, to be strictly real and within the unit circle. This indicates neither method should have any problems with instability or unwanted oscillations for any value of h.

*3.4.3 Study C*

By increasing the feedback capacitance $C_3$ to 100, a very stiff system results. Four different spectra are plotted in Figure 3.6. While both methods (Figures 3.6a and 3.6b) still remain stable for all h, the pure predictor method (gamma = 0.0) has a complex spectrum for h > 2.8. Increasing gamma to 1.5 completely removes the complex components of the predictor method (Figure 3.6c). The magnitude of the complex component reaches a maximum at h = 25.0 and in this case it is seen in Figure 3.7 that the pure predictor method (i.e., gamma = 0.0) has an oscillatory component in the solution that is not present using the other methods. As expected, increasing gamma to 1.5 will remove these oscillations. However, it was found experimentally that a gamma as small as 0.05 was enough to remove the oscillations and allowed a more accurate solution (Figure 3.8). The root locus plot for gamma = 0.05 (Figure 3.6d) shows that in this case the maximum magnitude of the complex part of the roots (now shifted to h = 12.5) was reduced by approximately 40 percent, from 0.5 to 0.3.

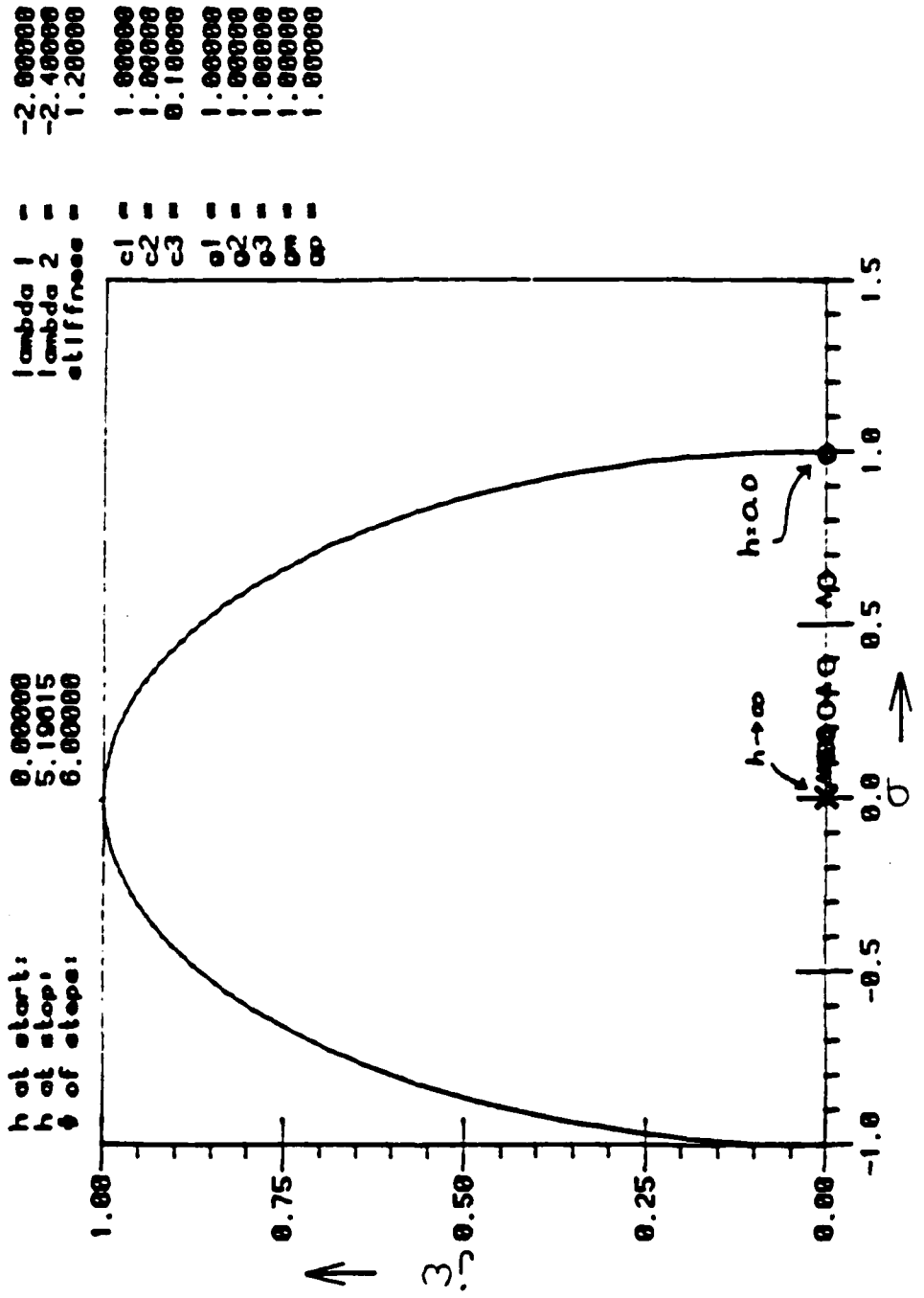



Figure 3.5a Standard Gauss-Seidel Spectrum for Study B.

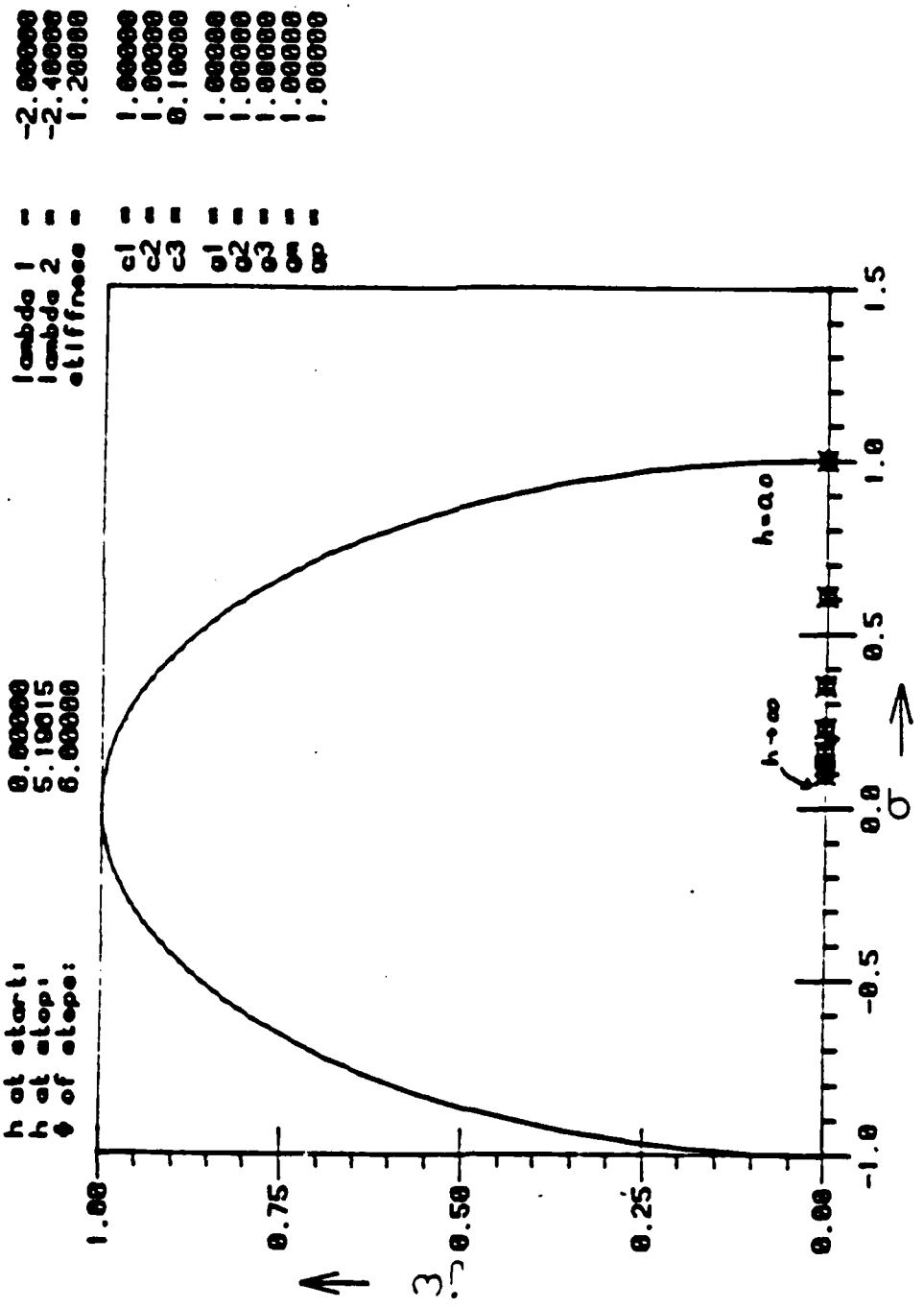



Figure 3.5b Predictor Gauss-Seidel Spectrum with gamma = 0.0 for Study B.

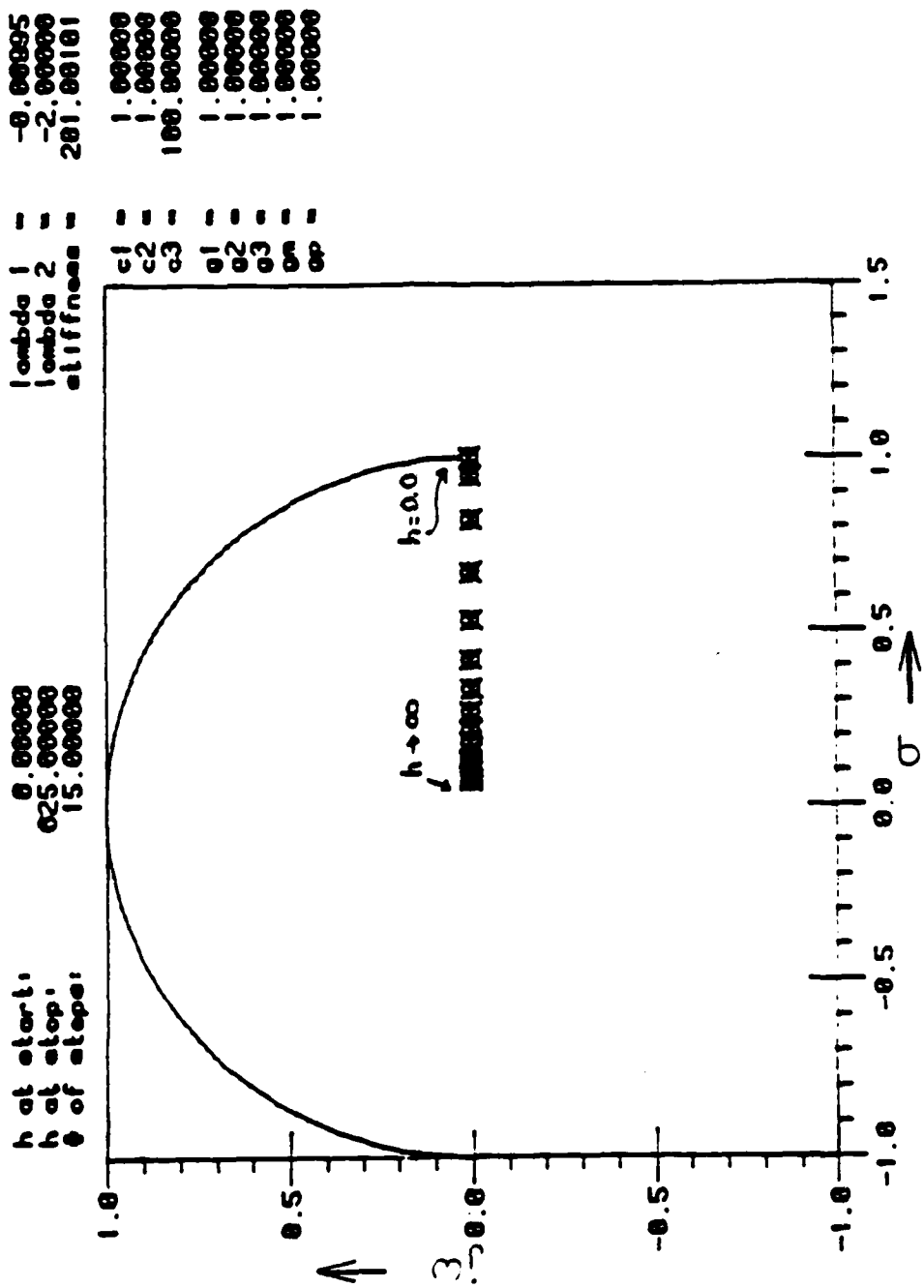



Figure 3.6a Standard Gauss-Seidel Spectrum for Study C.

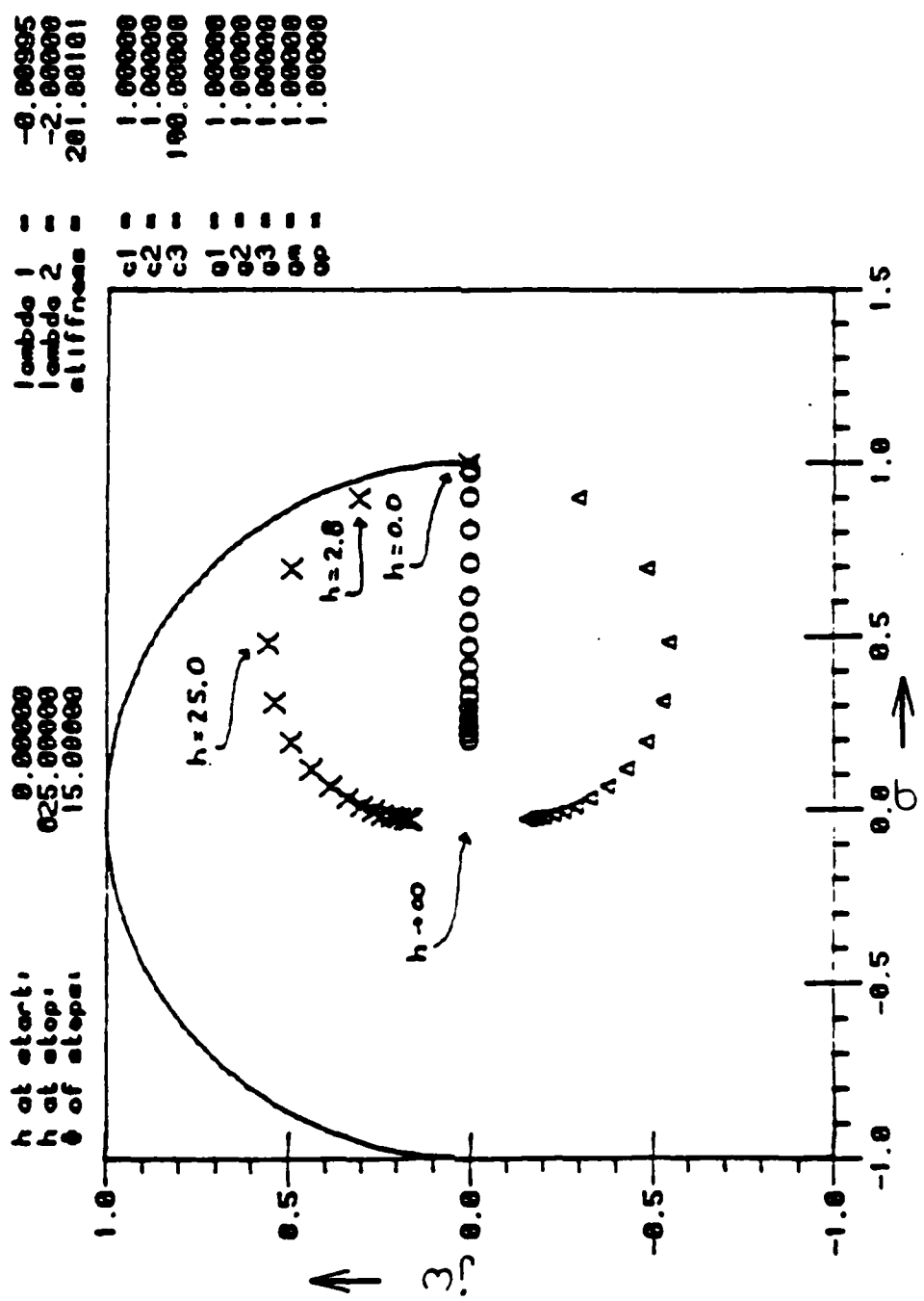



Figure 3.6b Predictor Gauss-Seidel Spectrum with gamma = 0.0 for Study C.

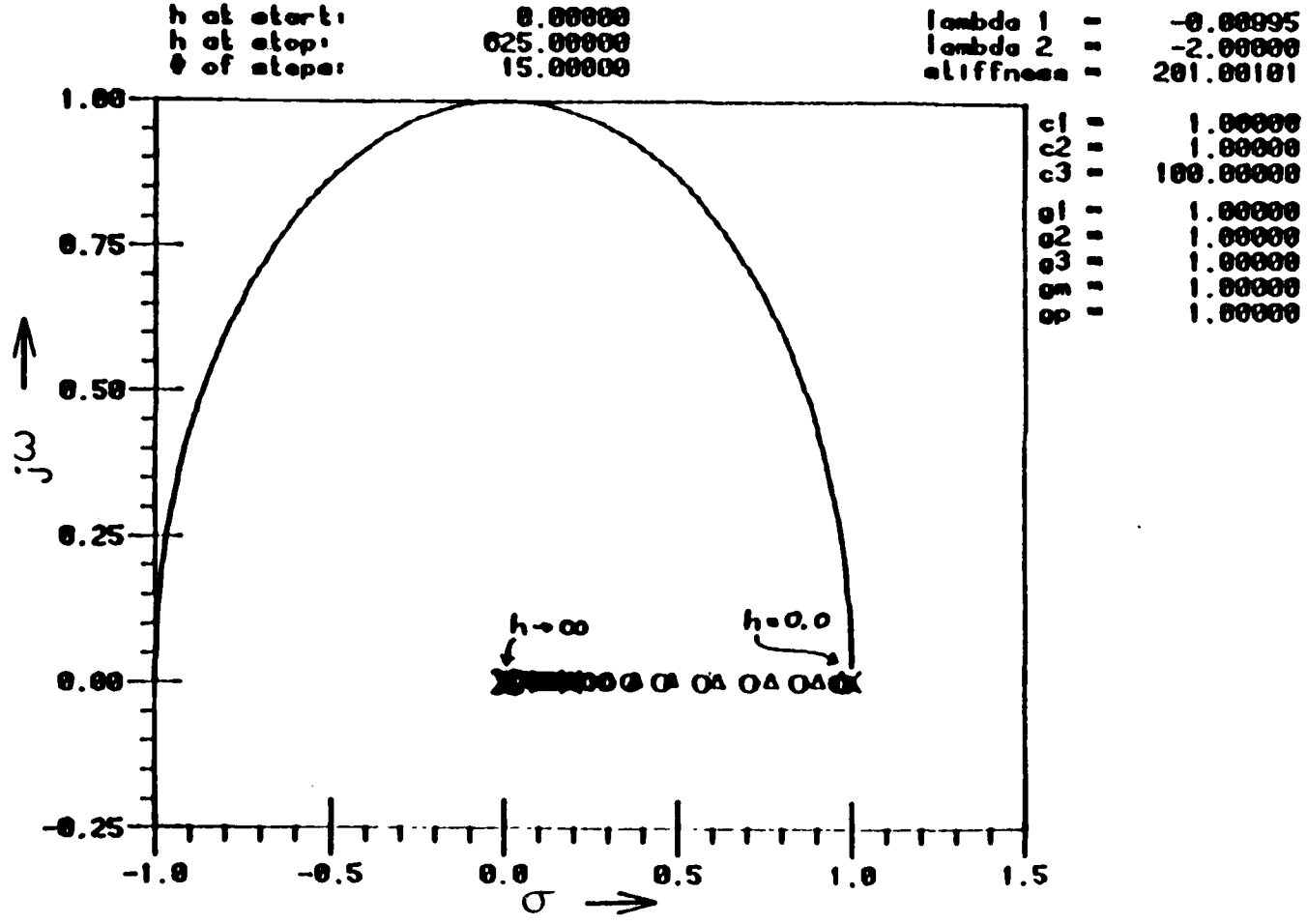



**Figure 3.6c** Predictor Gauss-Seidel Spectrum with gamma = 1.5 for Study C.

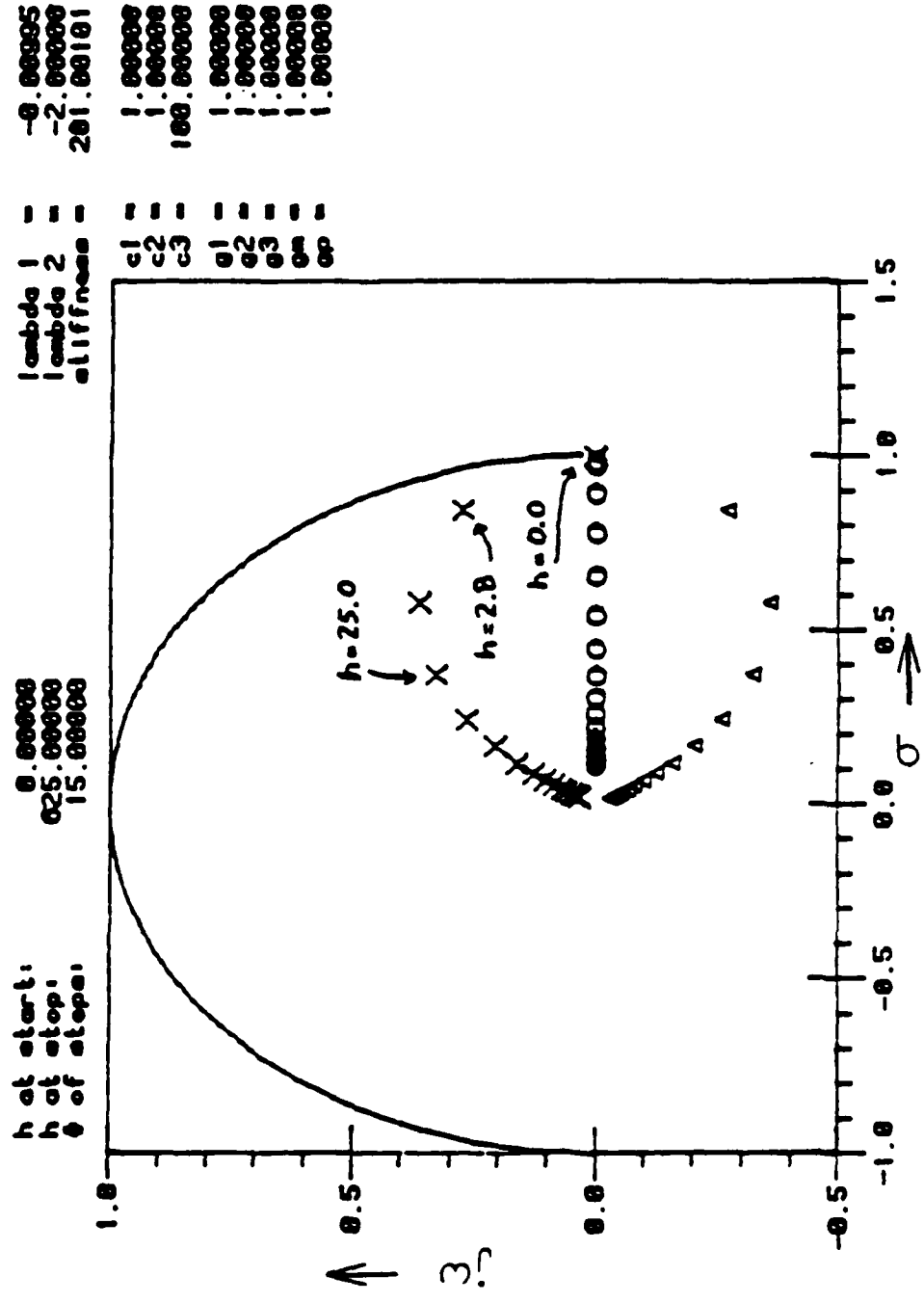



Figure 3.6d Predictor Gauss-Seidel Spectrum with gamma = 0.05 for Study C.

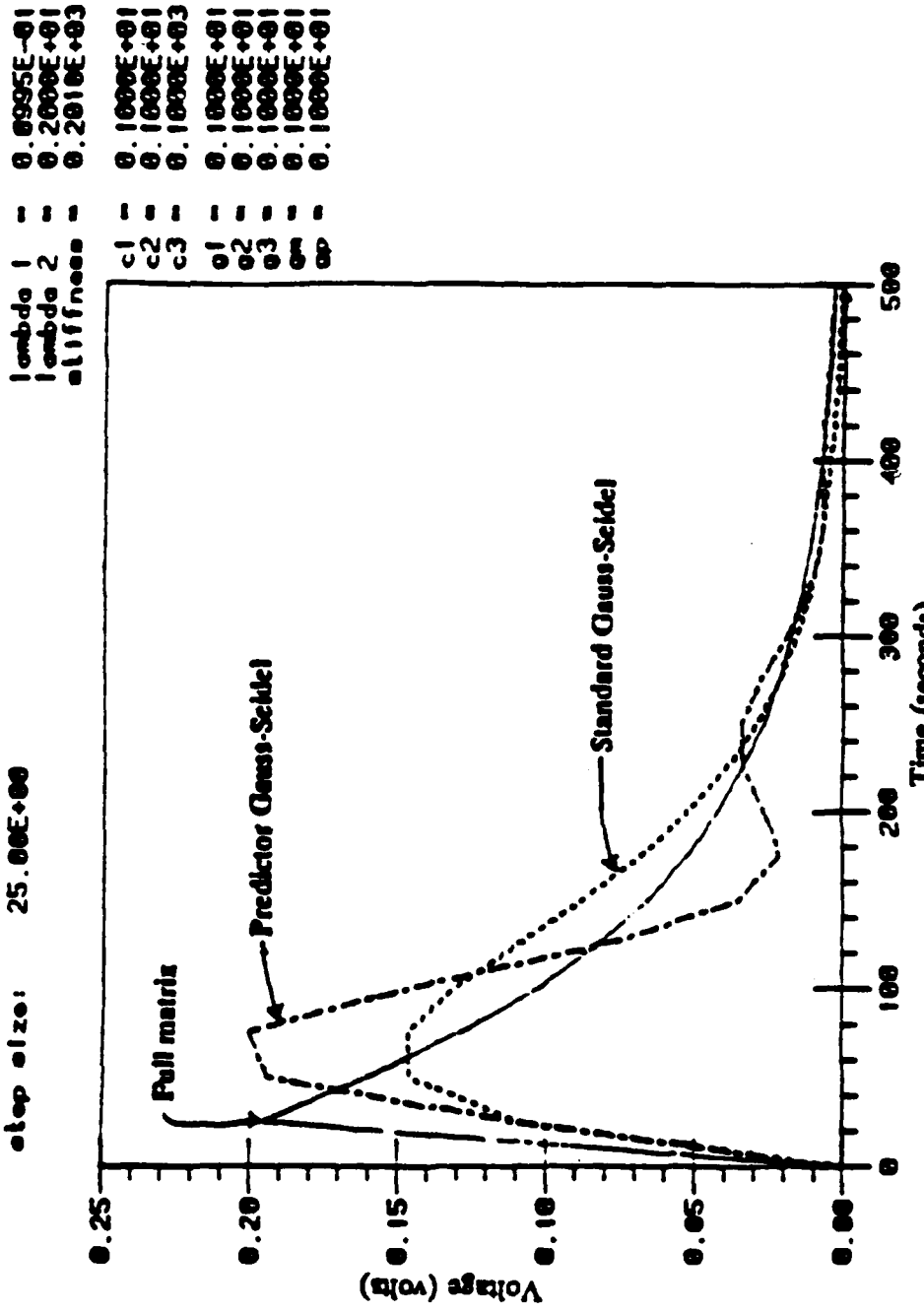



Figure 3.7 Node 2 output voltage with step size h = 25.0 and gamma = 0.0 for Study C.

with the solution generated by a full matrix approach, we see in this case that all three



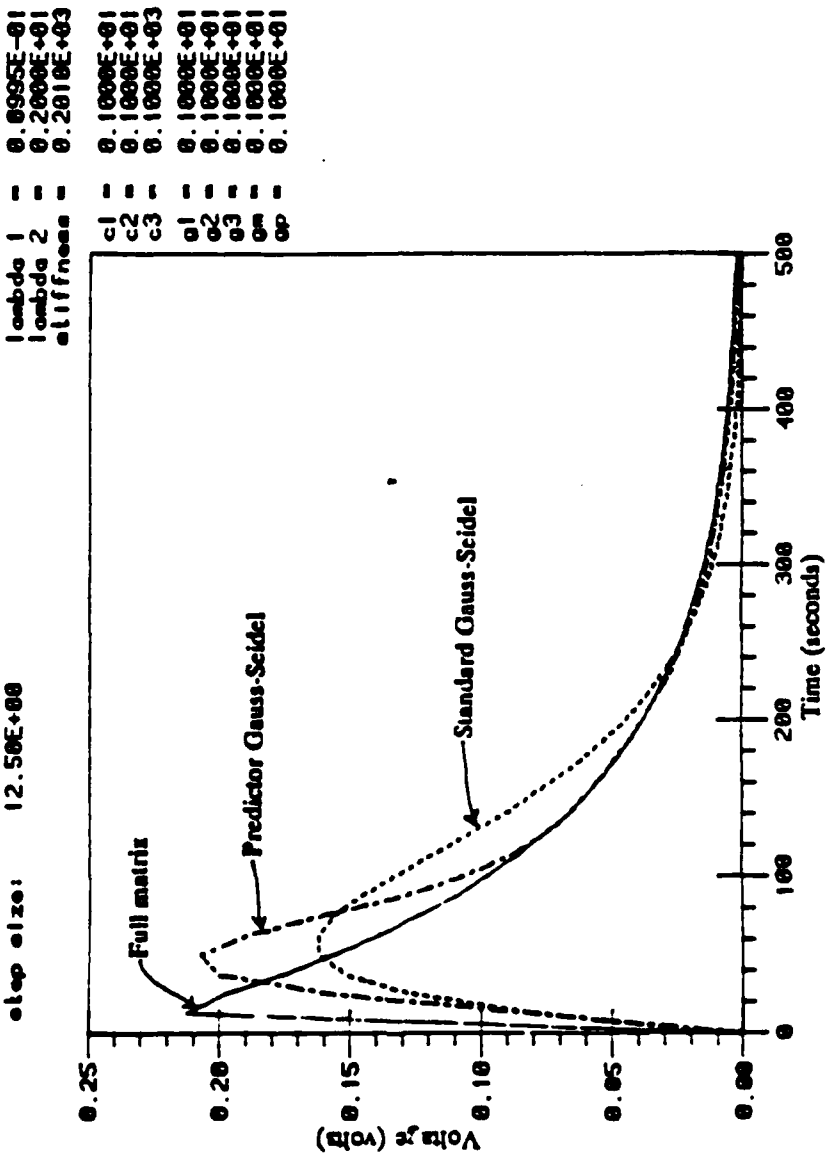



Figure 3.8 Node 2 output voltage with step size h = 12.5 and gamma = 0.05 for Study C.

Since the eigenvalues of the test circuit are strictly real (-0.1 and -2.0), the exact solution will not have any oscillatory components. Therefore, any oscillations in a computed solution are due to the numerical technique and not the physical circuit. By carefully choosing the step size, unwanted numerically generated oscillations can be observed. However, it should be noted that the step size necessary was much larger than either of the eigenvalues, and study A has already demonstrated that a step size smaller than the eigenvalues can be desirable for accuracy when an input to the circuit changes rapidly. This study also clearly shows that the existence of complex roots in the spectrum of the companion matrix is not enough to indicate oscillations are present in the computed solution. Since, in general, a step size larger than any of the eigenvalues may not generate an accurate solution in response to a rapidly changing input, this indicates that these unwanted oscillations are probably not a major factor in generating accurate solutions.

*3.4.4* *Study D*

In this case, with a very large feedback term represented by $g_p = 100$, we find that both the standard Gauss-Seidel (Figure 3.9a) and the predictor Gauss-Seidel (gamma = 0.0) (Figure 3.9b) methods become unstable for h within a finite interval. For this very stiff system with eigenvalues of -0.037 and -35.629, the solution would normally be obtained by starting with a very small step size to catch initial transients and then increasing the step size so that the rest of the interval would not require too much computation time. However, the standard method is unstable for $1 < h < 16$ while the pure predictor method is unstable for $1 < h < 81$. If an automatic time step control scheme increased the step size too much (i.e., $h > 1.0$), then neither method would be satisfactory.

For completeness, the spectrum of the predictor method with gamma = 2.0 is included as Figure 3.9c to demonstrate that the modified predictor method has properties in between those of the standard and the pure predictor methods.

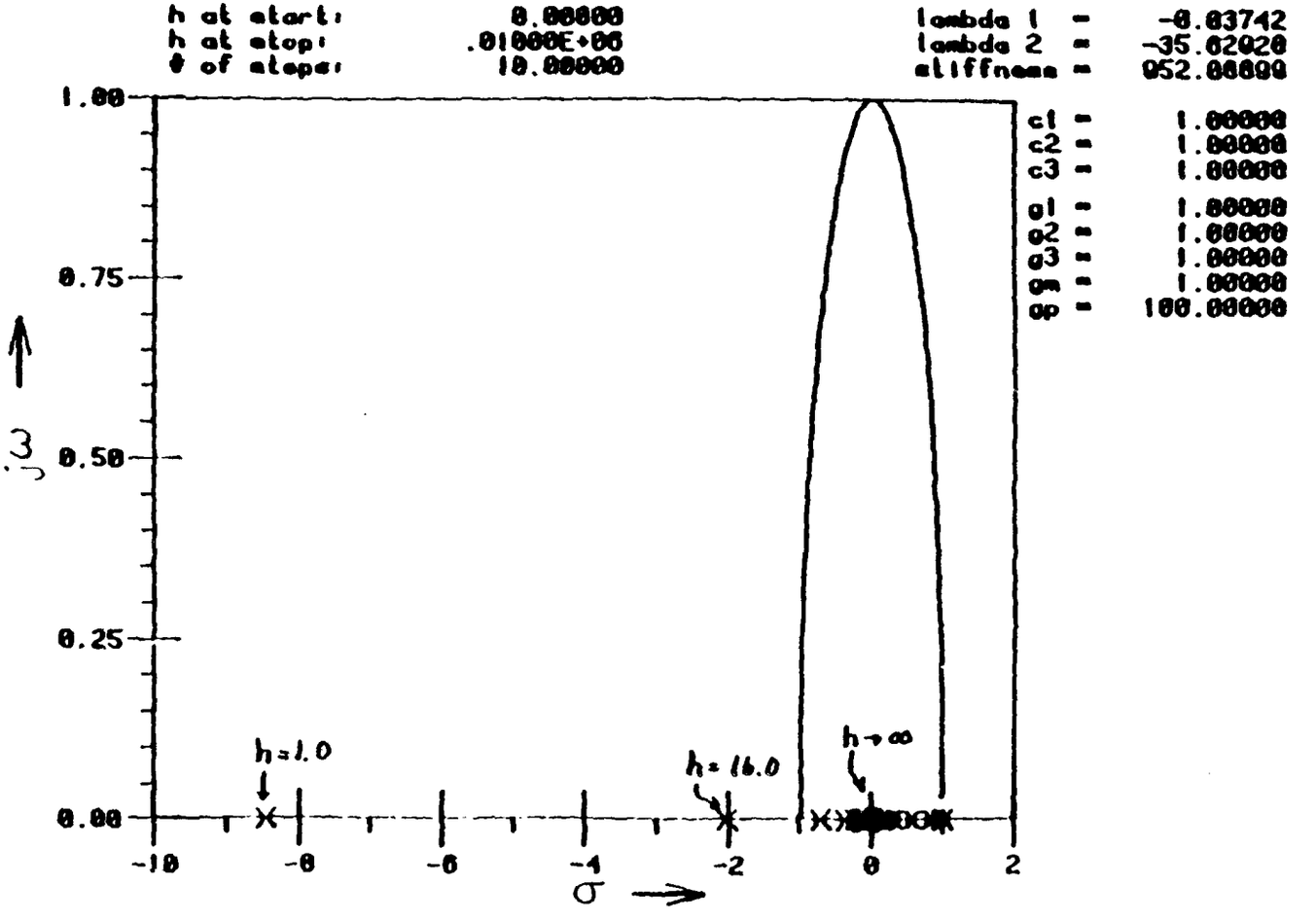



Figure 3.9c Predictor Gauss-Seidel Spectrum with gamma = 2.0 for Study D.

### 3.4.5 *Study E*

This case demonstrates properties of the methods when a very large transconductance is present. A simple exchange of the values for $g_p$ and $g_m$ yields a circuit with exactly the same eigenvalues as in study D. While the predictor method with gamma = 0.0 (Figure 3.10a) still has a finite interval where it is unstable, the standard Gauss-Seidel method (Figure 3.10b) is stable for all positive h. Comparing the C and G matrices for this test and for the previous test, we find that the only real difference is that the one element has moved to form a lower triangular instead of an upper triangular G matrix. To make the modified predictor method stable for all h, gamma is set to 5.0, as in Figure 3.10c.

An accuracy test with h = 81 (Figure 3.11) shows that the pure predictor method can oscillate around the full matrix solution when the step size is much larger than any of the eigenvalues. In order to use the predictor method with step size h = 1.0, gamma must be nonzero. If gamma = 5.0 as in the spectrum plot of Figure 3.10c, the accuracy plot of Figure 3.12 results. In this case (h = 1.0), neither the standard or the predictor Gauss-Seidel (gamma = 5.0) methods are very close to the full matrix approach. If the step size is reduced to 0.1, as in Figure 3.13, the predictor method comes much closer to the full matrix solution, while the standard Gauss-Seidel method continues to differ greatly. If, in addition to reducing the step size to 0.1, gamma is reduced from 5.0 to 0.0, then the pure predictor method is found to follow the full matrix solution almost exactly (Figure 3.14).

### 3.4.6 *Studies F and G*

Both of these cases lack the feedback capacitor, $c_3$, and have the same conductance matrix, G. The only difference is an interchange of the input and output capacitances, $c_1$ and $c_2$. This results in both circuits having identical eigenvalues and the major difference being a scale factor for each row of the A matrix. A comparison of the spectrum plots for study F (Figure 3.15) and for study G (Figure 3.16) shows that the results are identical when

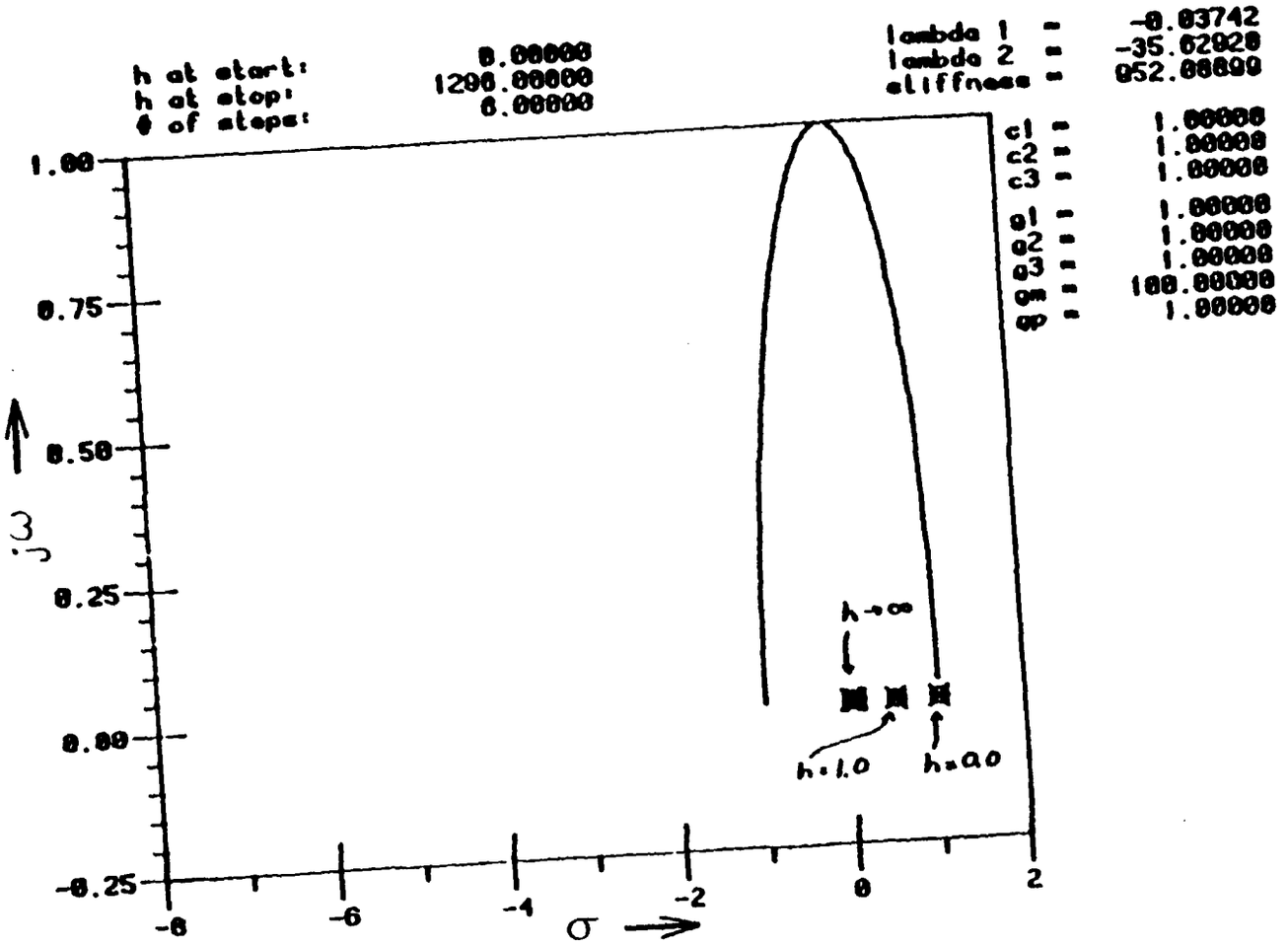



Figure 3.10a Standard Gauss-Seidel Spectrum for Study E.

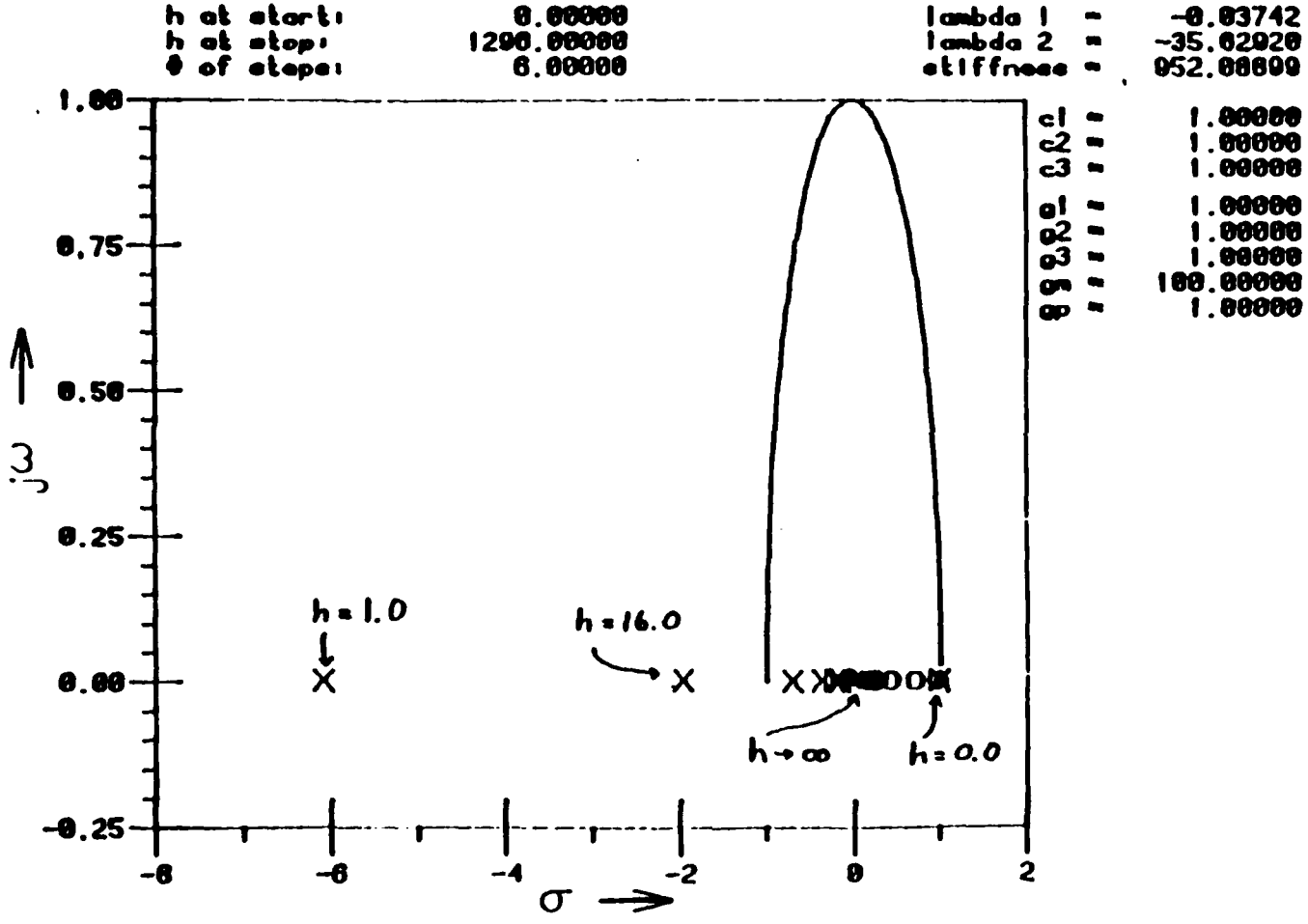



**Figure 3.10b** Predictor Gauss-Seidel Spectrum with gamma = 0.0 for Study E.

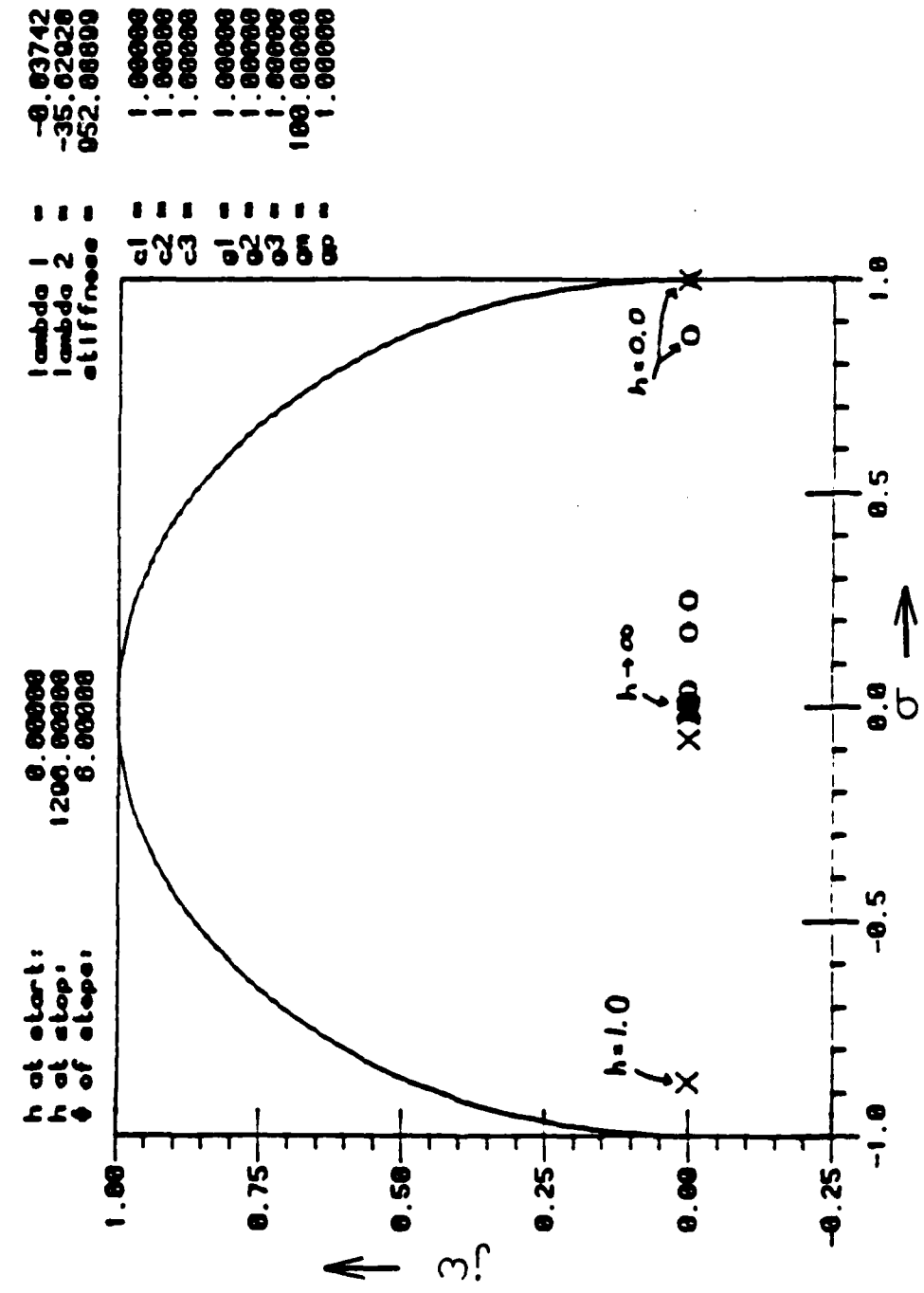



Figure 3.10c Predictor Gauss-Seidel Spectrum with gamma = 5.0 for Study E.

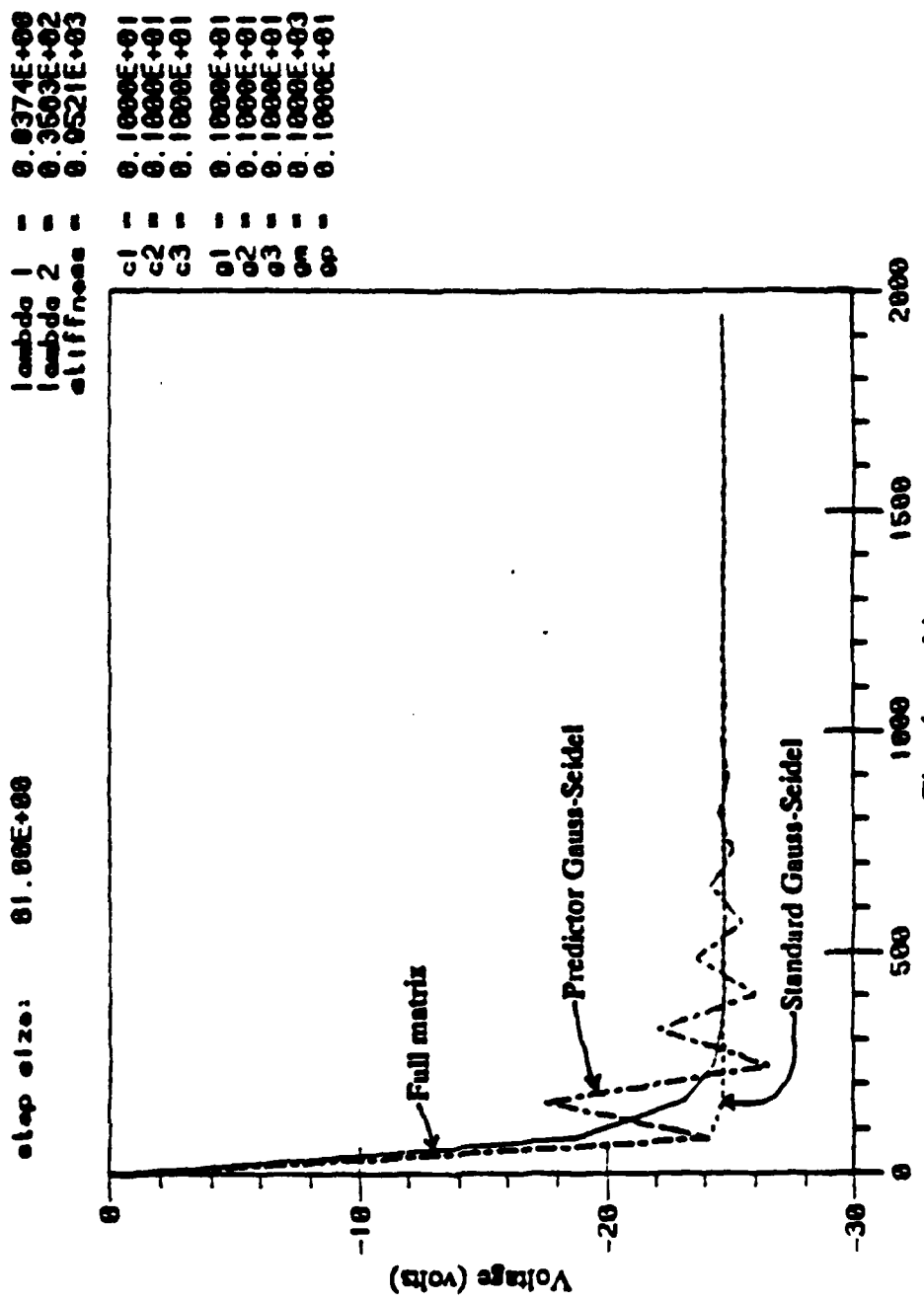



Figure 3.11 Node 2 output voltage with step size h = 81.0 and gamma = 0.0 for Study E.

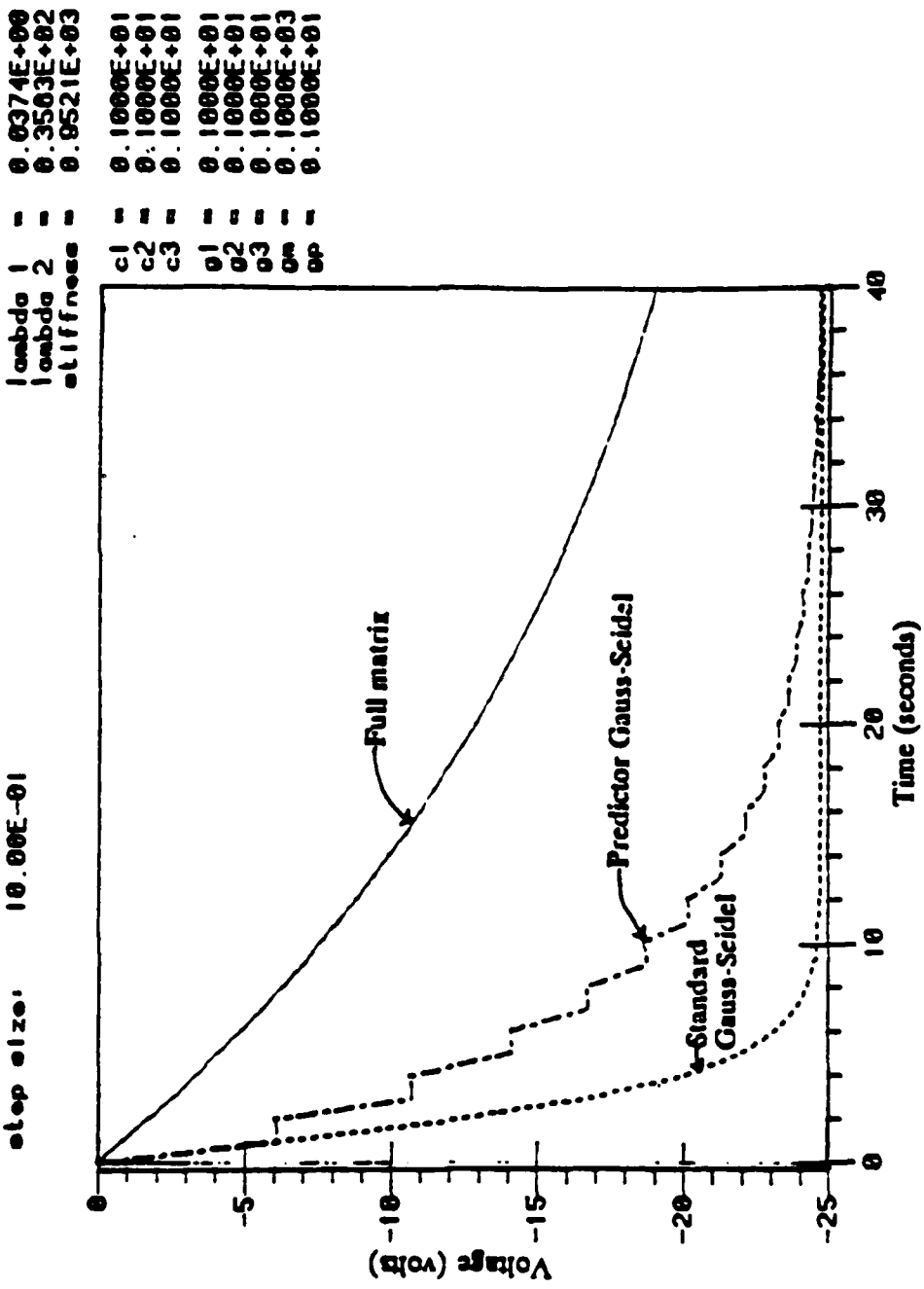



Figure 3.12 Node 2 output voltage with step size h = 1.0 and gamma = 5.0 for Study E.

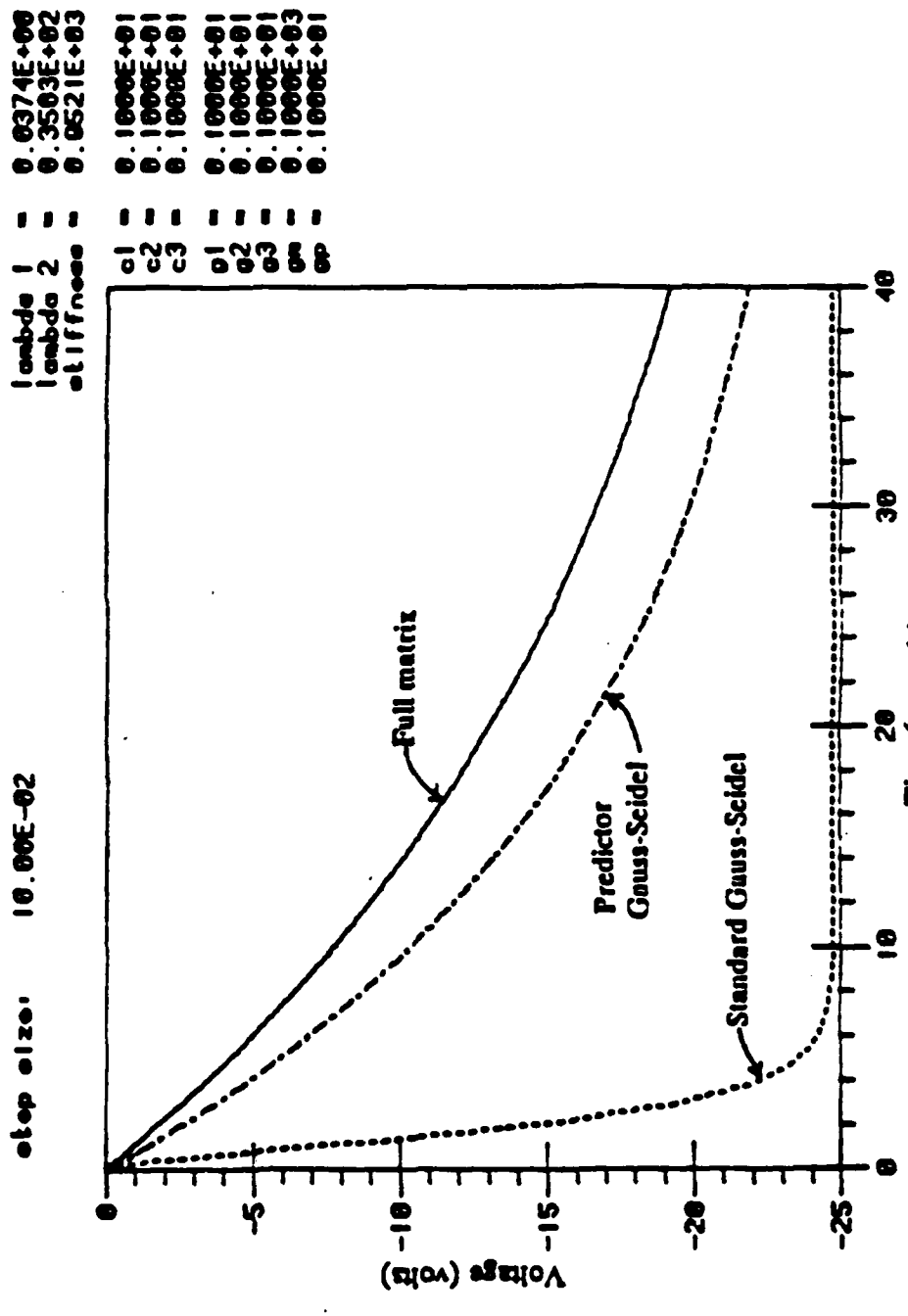



Figure 3.13 Node 2 output voltage with step size h = 0.1 and gamma = 5.0 for Study E.

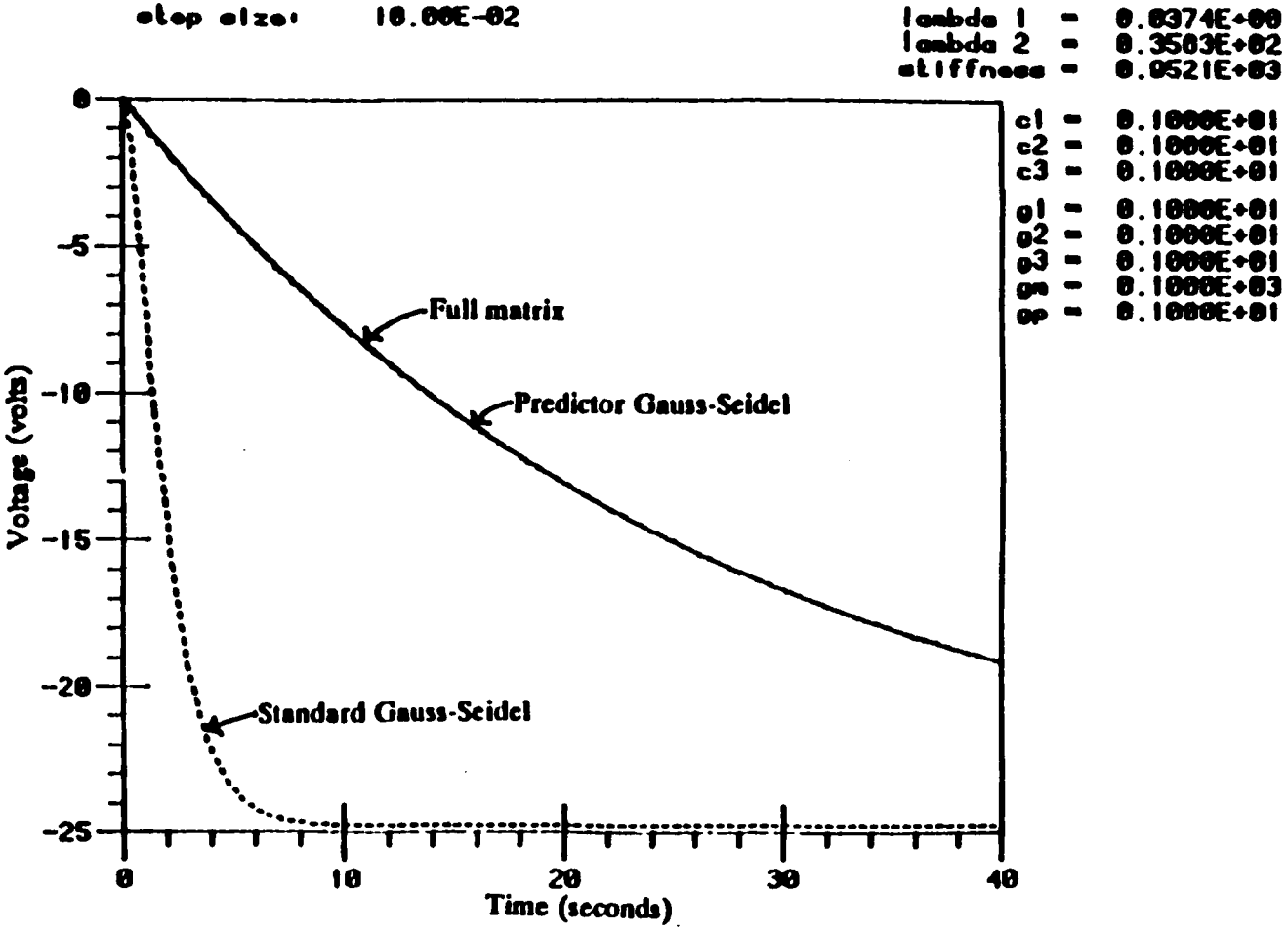



Figure 3.14 Node 2 output voltage with step size h = 0.1 and gamma = 0.0 for Study E.

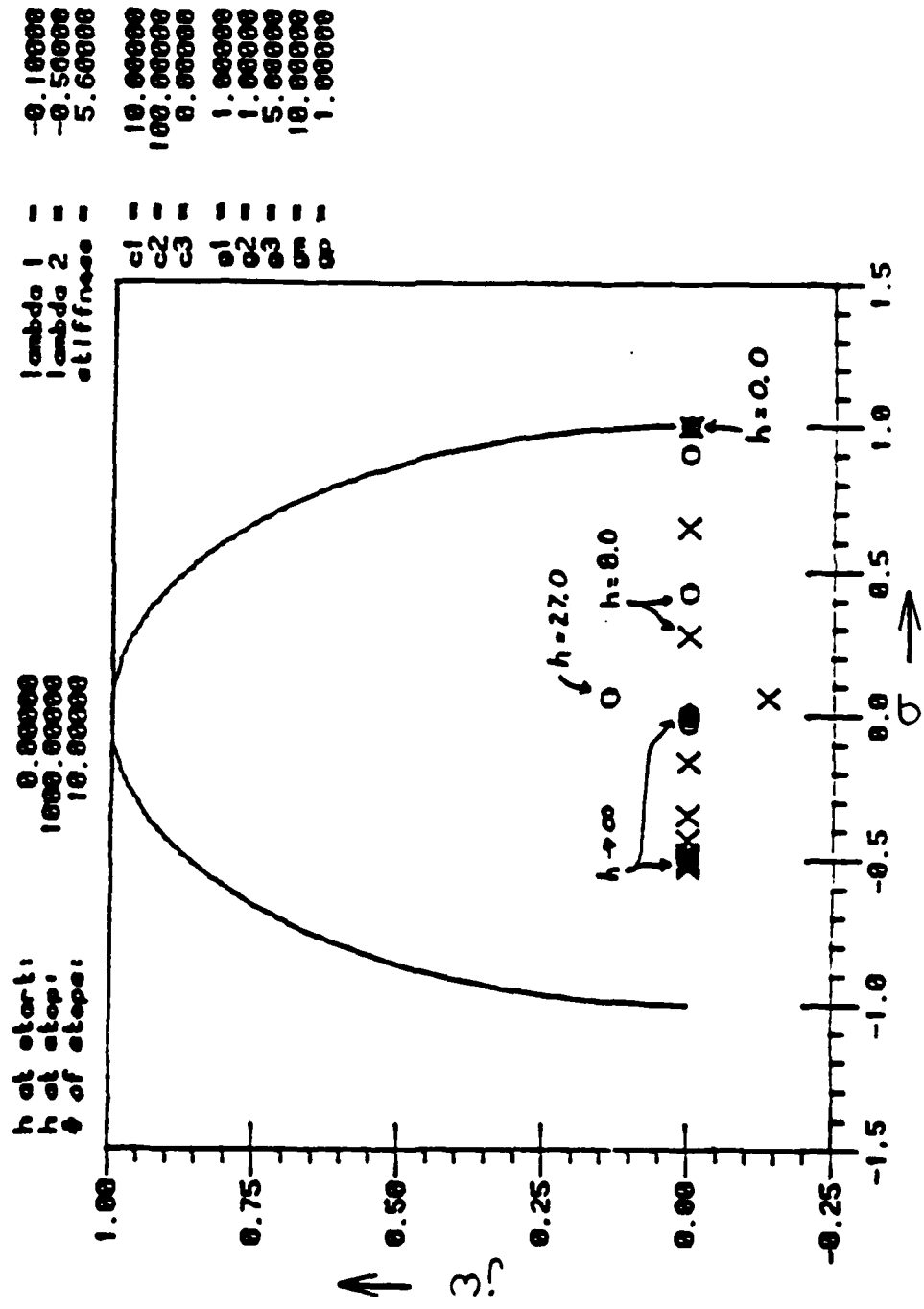



Figure 3.15a Standard Gauss-Seidel Spectrum for Study F.

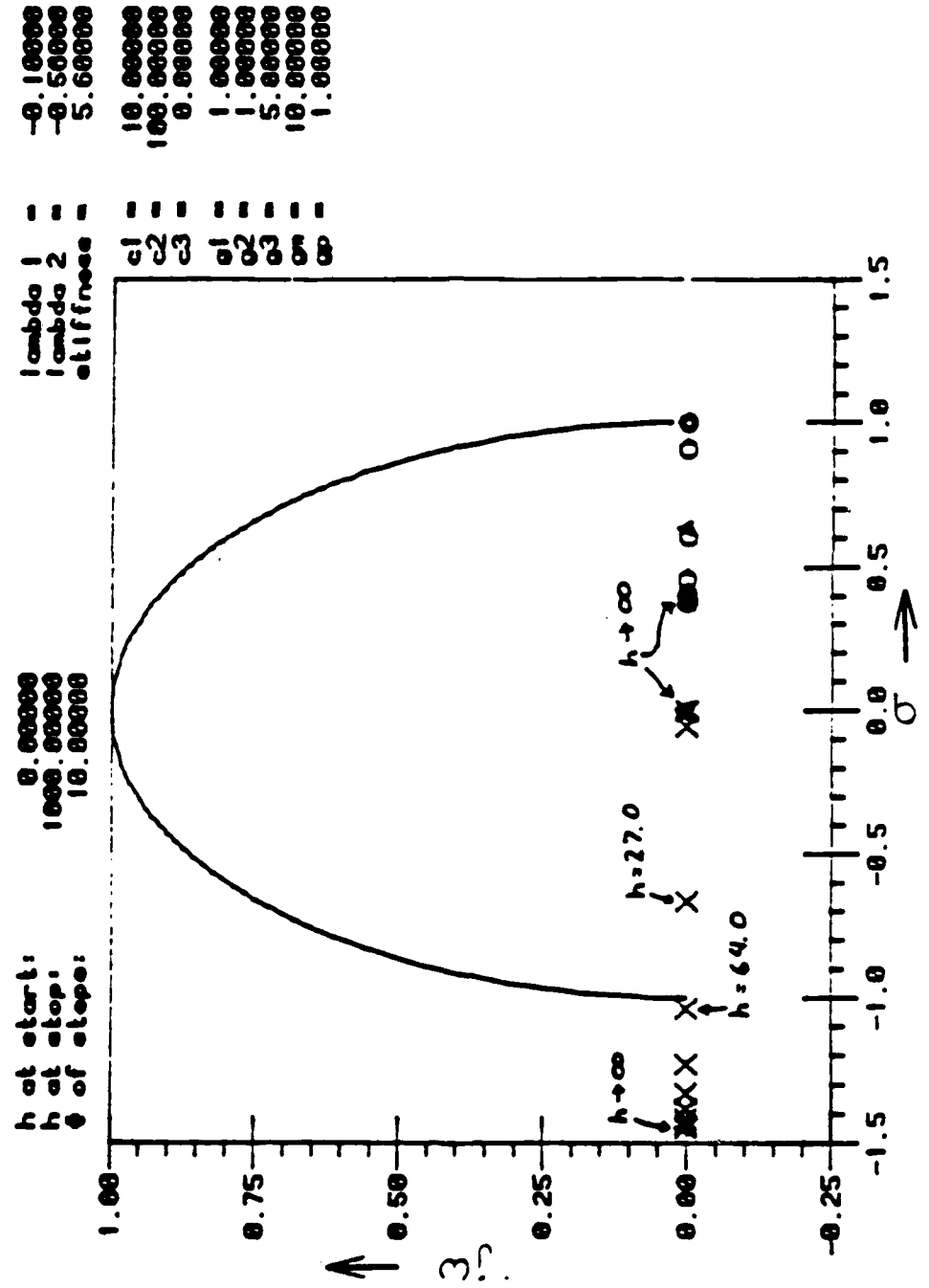



Figure 3.15b Predictor Gauss-Seidel Spectrum with gamma = 0.0 for Study F.

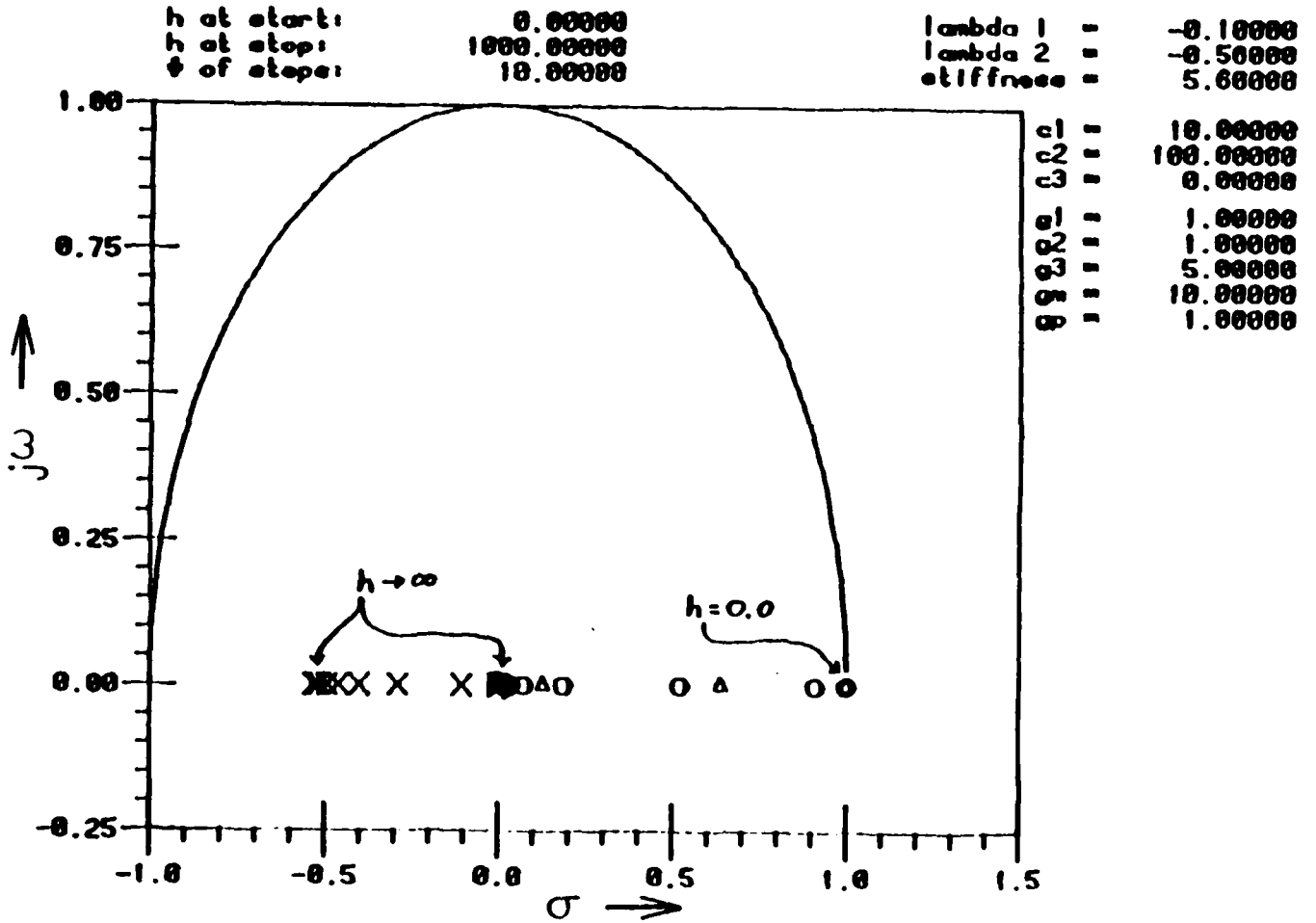



Figure 3.15c Predictor Gauss-Seidel Spectrum with gamma = 0.25 for Study F.

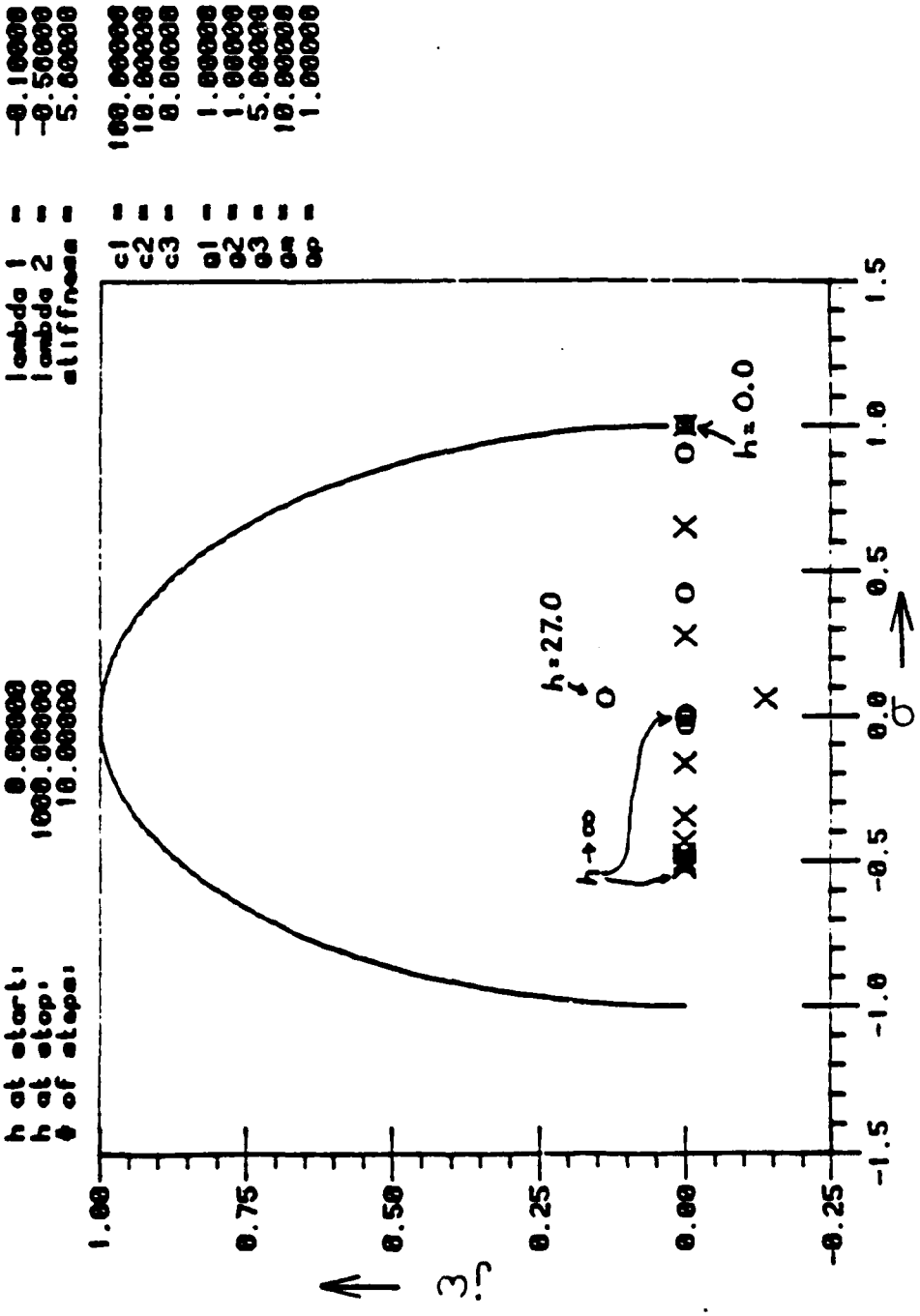



Figure 3.16a Standard Gauss-Seidel Spectrum for Study G.

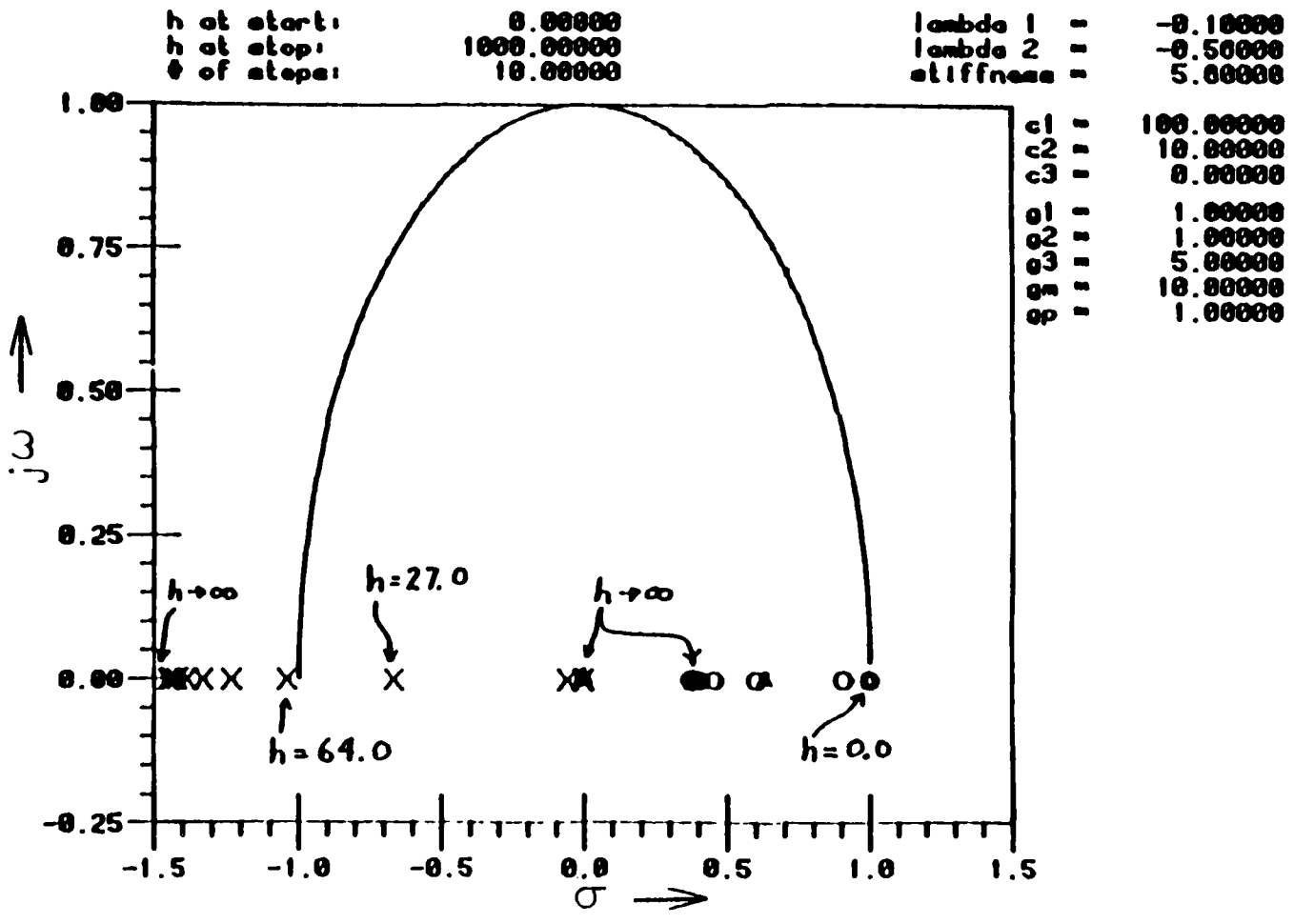



Figure 3.16b Predictor Gauss-Seidel Spectrum with gamma = 0.0 for Study G.

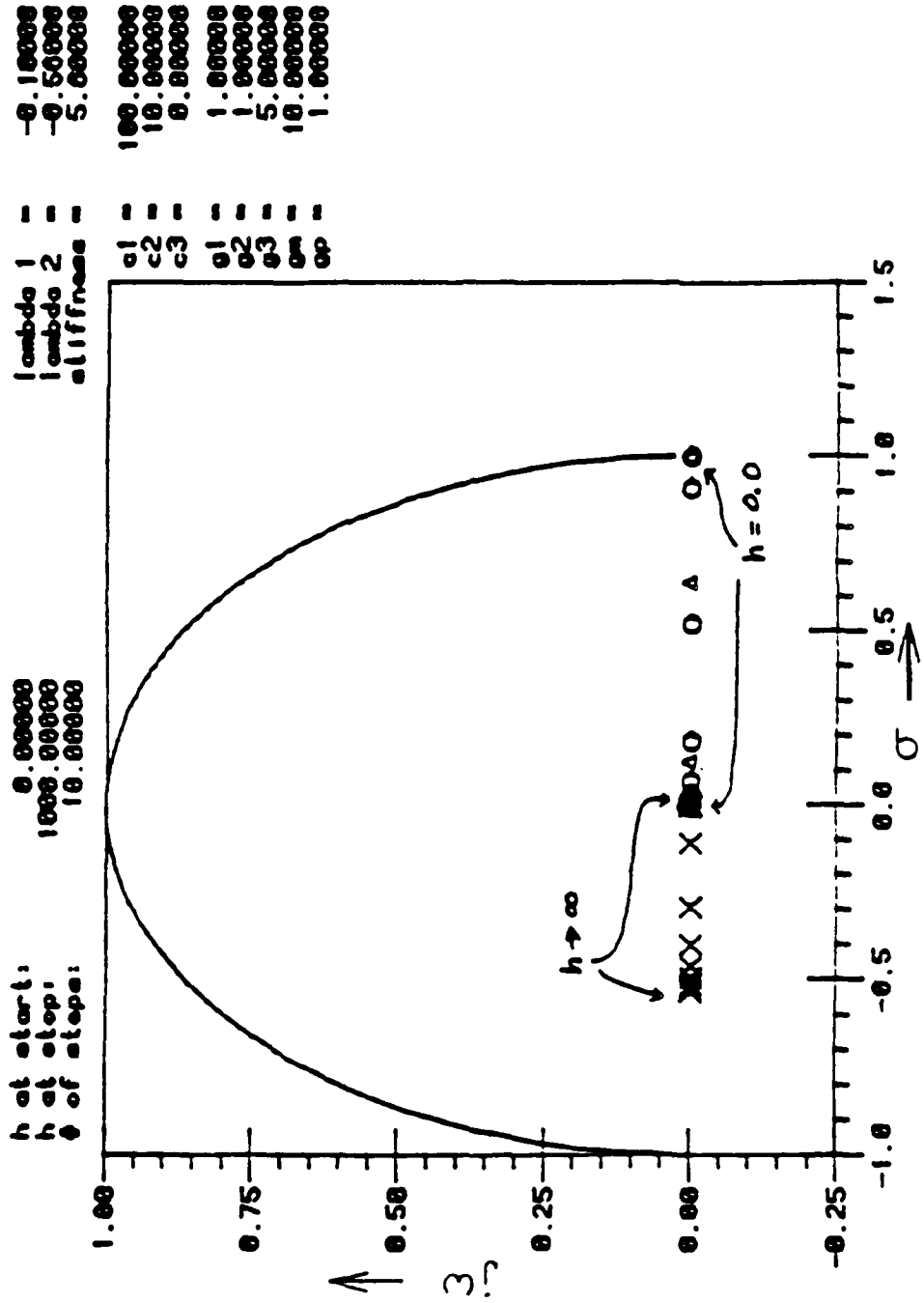



Figure 3.16c Predictor Gauss-Seidel Spectrum with gamma = 0.9 for Study G.

a specific method is compared to the same method. One interesting result of these studies is that even though both the capacitance and conductance matrices are diagonally dominant (i.e., for each row, the sum of the off-diagonal elements is less than the absolute value of the diagonal element [14]), the predictor Gauss-Seidel method is unstable for all h > 64.0, while the standard Gauss-Seidel method is stable for all h. An accuracy plot (Figure 3.17) shows that for h comparable to the eigenvalues, both the standard and predictor Gauss-Seidel methods have accuracy very close to that of a full matrix solution.

The studies conducted so far have been chosen to illustrate behavior with respect to specific component values in the test circuit. The next three studies are included to allow further comparison based upon the structure of the capacitance and conductance matrices.

### *3.4.7 Studies H, J, and K*

Study H demonstrates behavior when all the elements in both the capacitance and conductance matrices are nonzero. As in study F, both the C and G matrices are diagonally dominant. A wider range of h for which the spectra are complex is observed in Figures 3.18a and 3.18b. The predictor method still is unstable for large enough h, while the standard method is stable for all h.

Study J has the same C and G matrices as study H, except that the upper hand element of the conductance matrix has been set to zero. This creates a lower triangular, diagonally dominant G matrix. In this example, Figures 3.19a and 3.19b show that both methods are stable for all h.

Study K again has the same C and G matrices as study H, except that this time the lower lefthand element has been set to zero. This yields an upper triangular conductance matrix. As in study J, both the standard and predictor Gauss-Seidel (gamma = 0.0) methods are stable for all h. (Figure 3.20.)

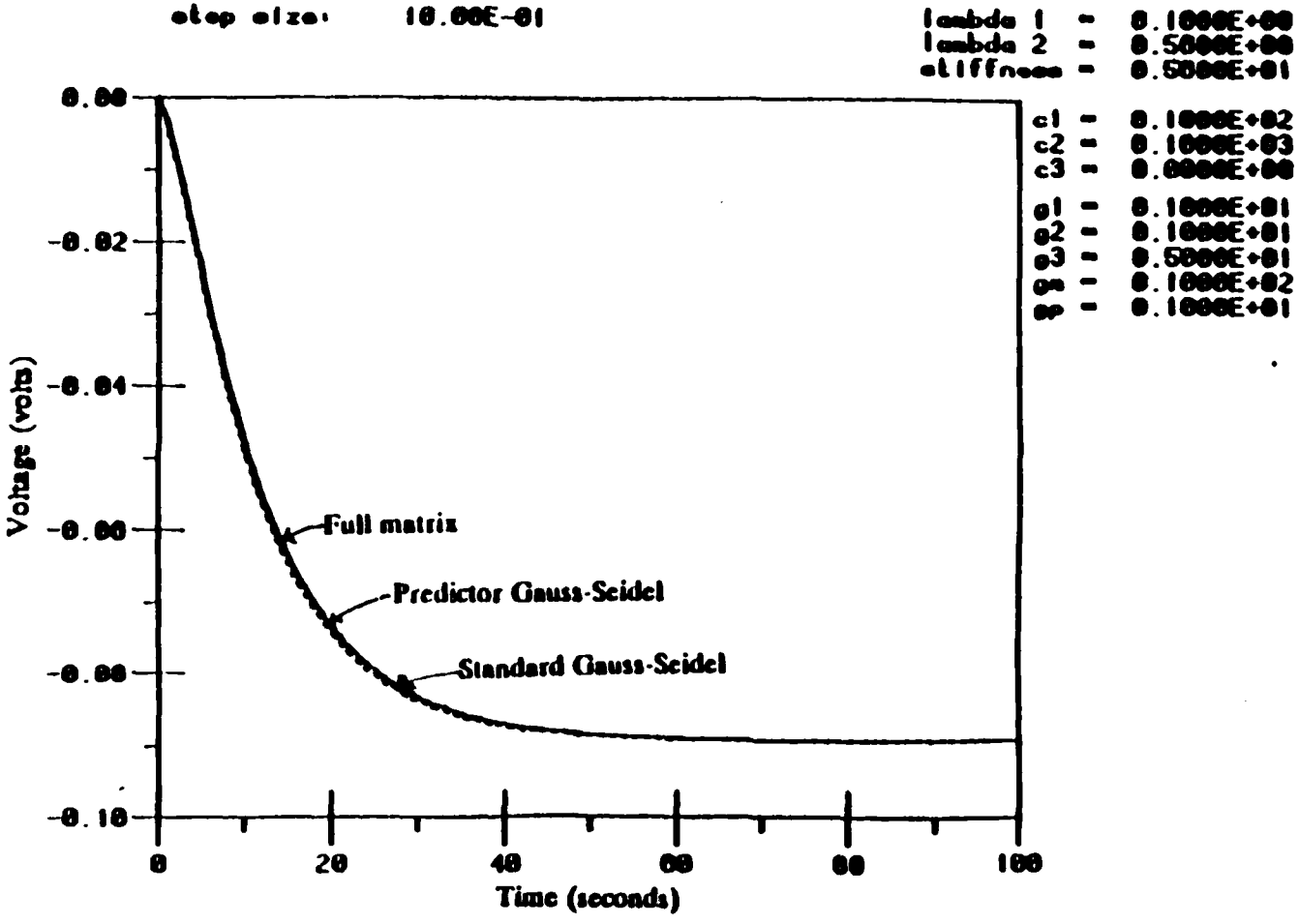



Figure 3.17 Node 2 output voltage with step size h = 1.0 and gamma = 0.0 for Study F.

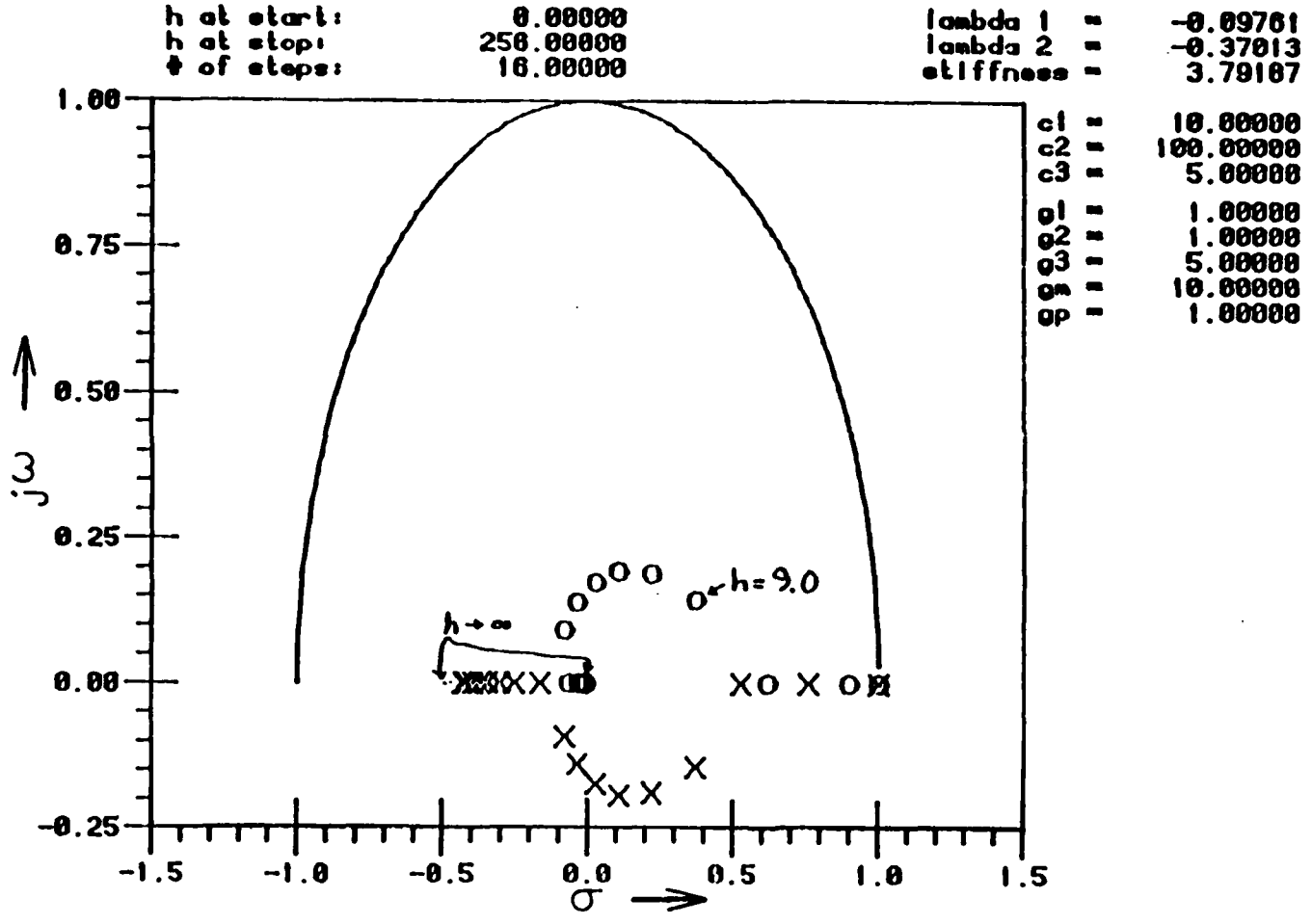



Figure 3.18a Standard Gauss-Seidel Spectrum for Study H.

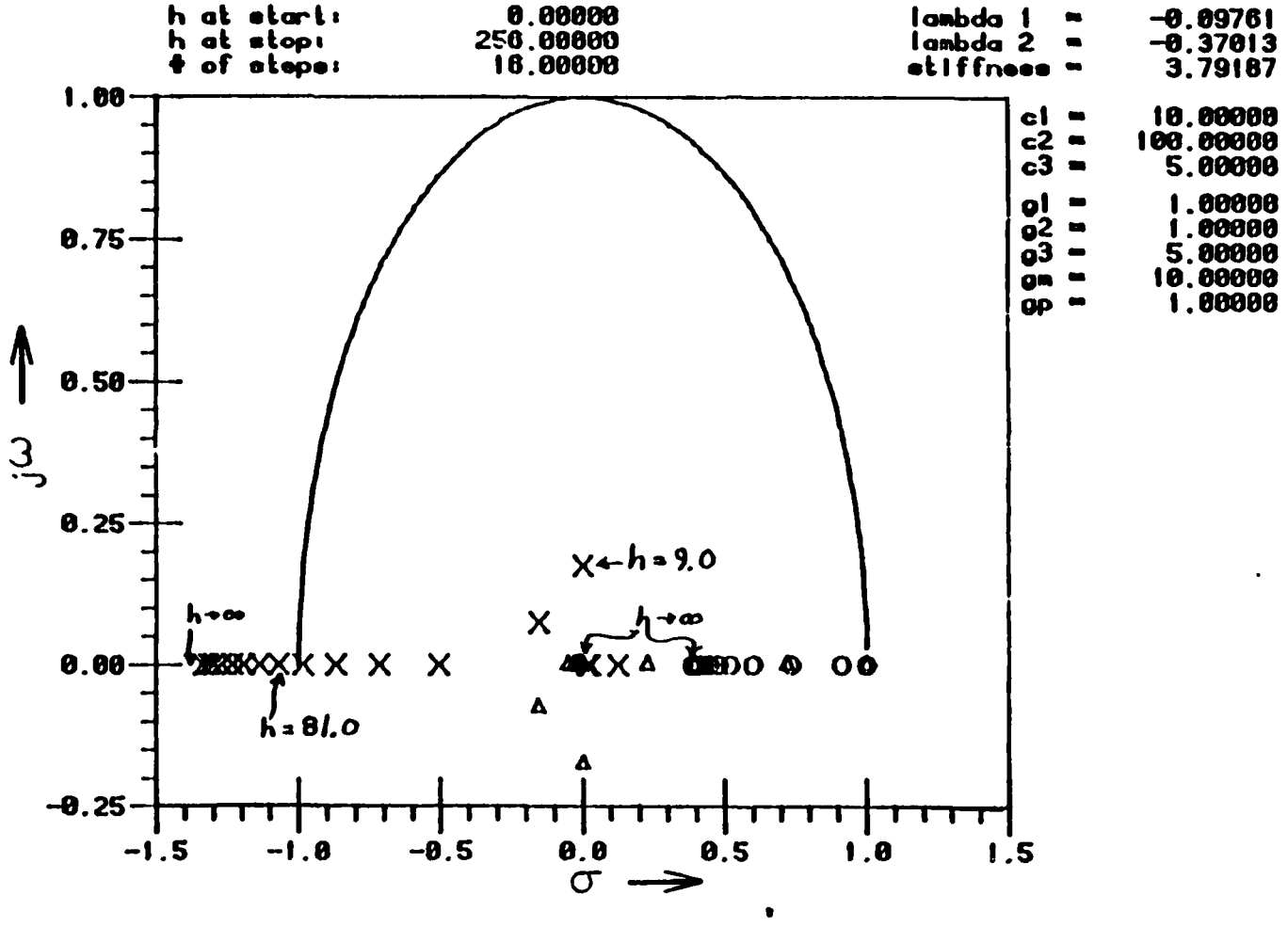



Figure 3.18b Predictor Gauss-Seidel Spectrum with gamma = 0.0 for Study II.

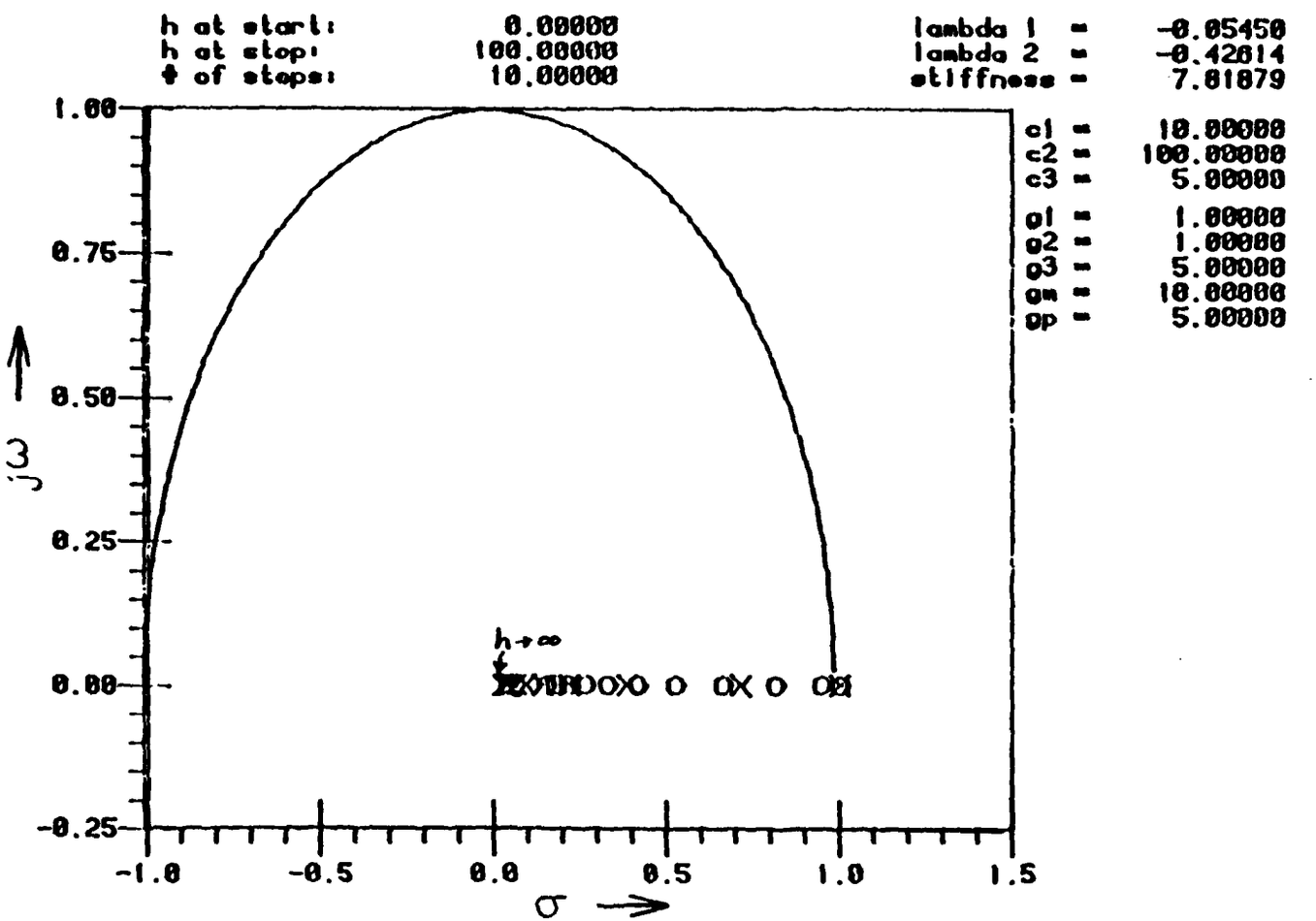



Figure 3.19a Standard Gauss-Seidel Spectrum for Study I.

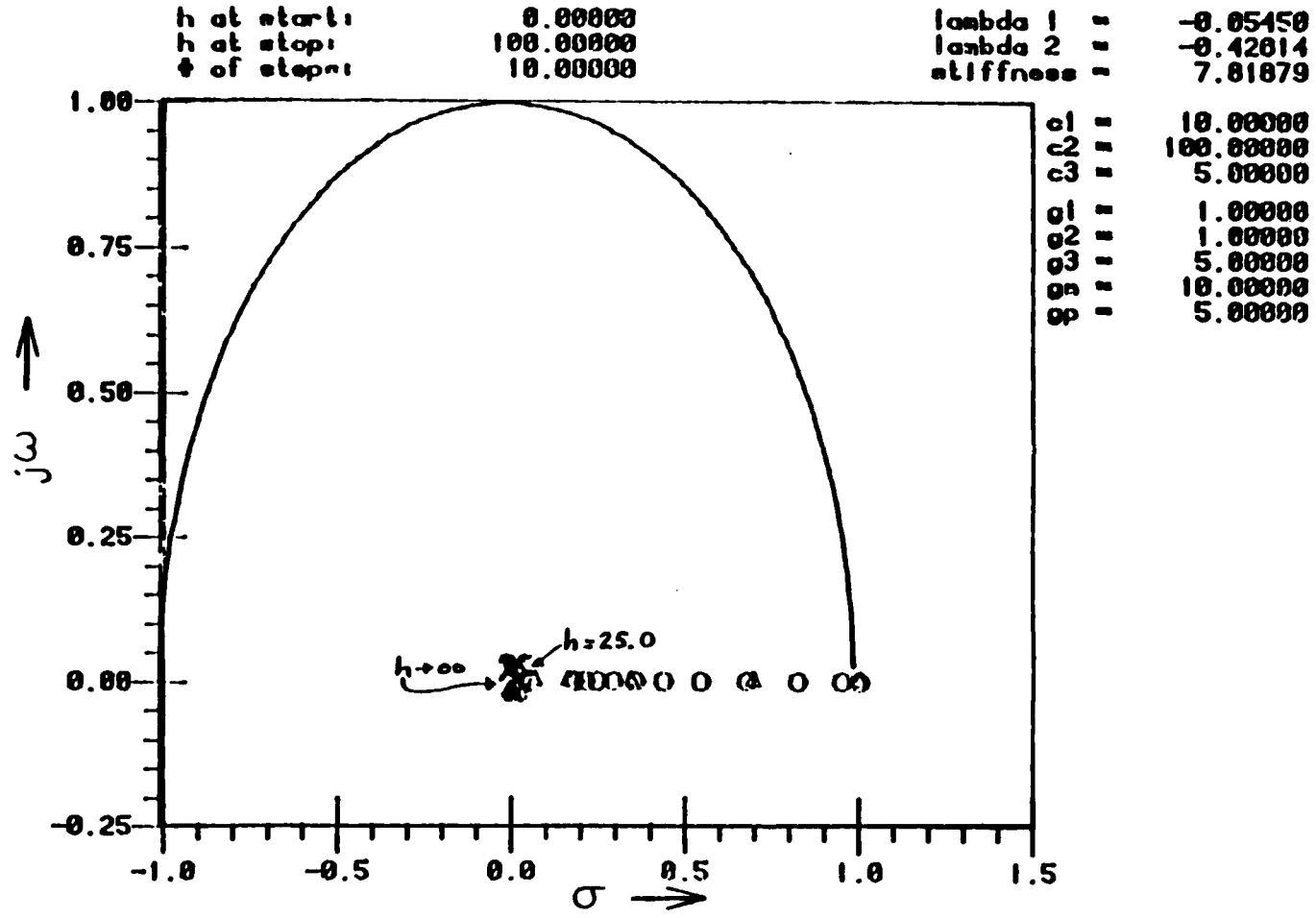



Figure 3.19b Predictor Gauss-Seidel Spectrum with gamma = 0.0 for Study J.

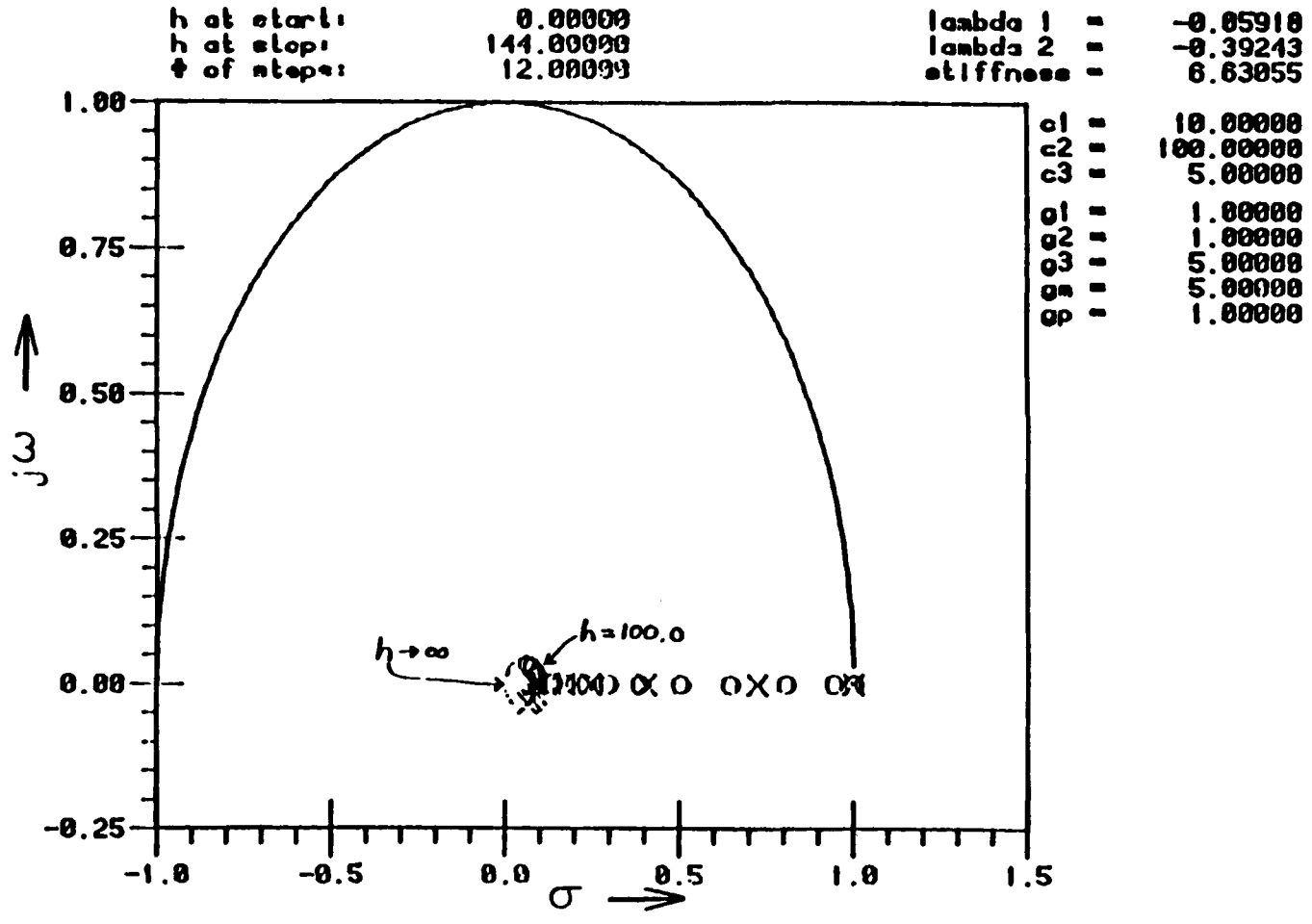



Figure 3.20a Standard Gauss-Seidel Spectrum for Study K.

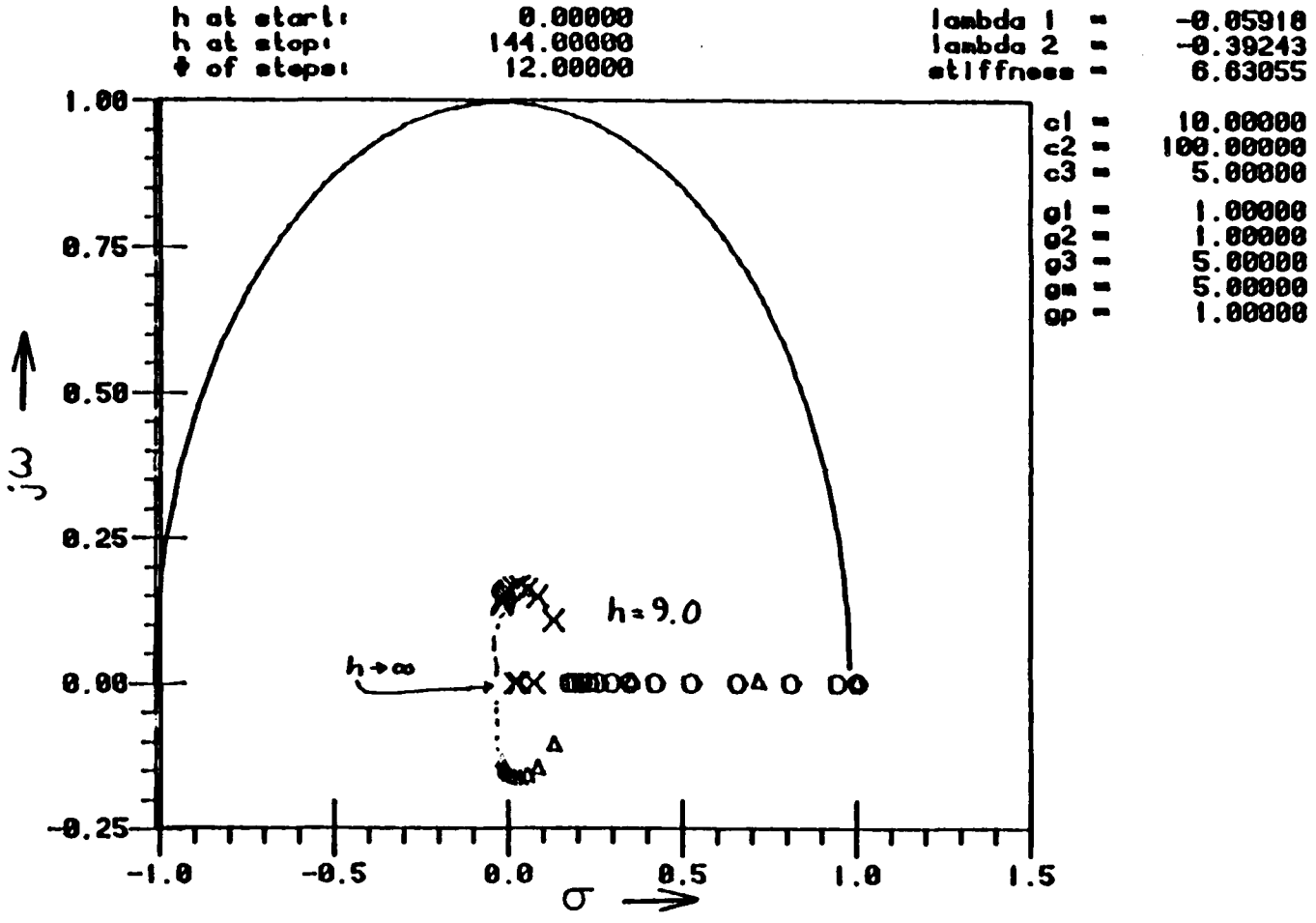



Figure 3.20b Predictor Gauss-Seidel Spectrum with gamma = 0.0 for Study K.

### 3.5 Discussion

The results of each of these ten studies are summarized in Figure 3.21. Included for each study are the capacitance (C), conductance (G), and $A = (-C^{-1}G)$ matrices, along with the eigenvalues of the test circuit. In the following discussion, the term "standard method" means "standard Gauss-Seidel method" and the term "predictor method" means "predictor Gauss-Seidel method."

A comparison of studies A, B, and C shows that the differences result from the difference in the capacitance matrix. By referring to Equation (3.6), it can be seen that the C matrix will always be symmetric (i.e., $C = C^T$)[14] and will be diagonally dominant as long as $c_1$ and $c_2$ are both nonzero. This indicates that if the G matrix is strictly diagonal, both the standard and predictor methods will be stable for all h. If the feedback capacitance, $c_3$, is very large in comparison to the other two capacitance then some undesirable oscillation can occur, but stability is maintained.

The next two studies, D and E, are the only two studies with conductance matrices that are not diagonally dominant. They are also the only two studies with a finite range of h where the predictor method is unstable. In every other study, if the predictor method is stable for some $h_m$, then the method is stable for all $h < h_m$. This stability property is also true for the standard method in study D. (The standard method is stable for all h in all studies except D.) Since the interval of h for which the methods are unstable includes a region between the value of the eigenvalues, it is likely that a fast and accurate solution would be difficult to achieve for study D with either method. (This is because a step size too large would either be in the unstable region or could cause large inaccuracies. But a small step size slows the speed of analysis.)

| Study | C matrix | G matrix | A matrix | eigenvalues |
|---|---|---|---|---|
| A | $\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$ | $\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ | $\frac{-1}{3}\begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}$ | -0.66, -2.0 |
| | Comments: Both standard and predictor methods are stable for all n. Predictor method has complex component in spectrum for n > 0.33, complex component reaches a maximum at n = 1.0 | | | |
| B | $\begin{bmatrix} 1.1 & -0.1 \\ -0.1 & 1.1 \end{bmatrix}$ | $\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ | $\frac{-1}{1.2}\begin{bmatrix} 2.2 & 2.0 \\ 2.0 & 2.2 \end{bmatrix}$ | -2.66, -2.0 |
| | Comments: Both standard and predictor methods are stable for all n. Both have strictly real spectra for all n. | | | |
| C | $\begin{bmatrix} 101 & -100 \\ -100 & 101 \end{bmatrix}$ | $\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ | $\frac{-1}{201}\begin{bmatrix} 202 & 200 \\ 200 & 202 \end{bmatrix}$ | -0.01, -2.0 |
| | Comments: Both standard and predictor methods are stable for all n. Spectrum of predictor method has complex components for n > 2.8, complex component reaches a maximum at n = 25.0 | | | |
| D | $\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$ | $\begin{bmatrix} 2 & 99 \\ 0 & 2 \end{bmatrix}$ | $\frac{-1}{3}\begin{bmatrix} 4 & 200 \\ 2 & 103 \end{bmatrix}$ | -0.04, -35.6 |
| | Comments: Both standard and predictor methods unstable for finite range of n, 1.0 < n < 16.0 for standard and 1.0 < n < 81.0 for predictor | | | |
| E | $\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$ | $\begin{bmatrix} 2 & 0 \\ 99 & 2 \end{bmatrix}$ | $\frac{-1}{3}\begin{bmatrix} 103 & 2 \\ 200 & 4 \end{bmatrix}$ | -0.04, -35.6 |
| | Comments: Standard method stable for all n, predictor method unstable for 1.0 < n < 16.0 | | | |
| F | $\begin{bmatrix} 10 & 0 \\ 0 & 100 \end{bmatrix}$ | $\begin{bmatrix} 6 & -4 \\ 0 & 6 \end{bmatrix}$ | $\frac{-1}{1000}\begin{bmatrix} 600 & -400 \\ 90 & 60 \end{bmatrix}$ | -0.10, -0.56 |
| | Comments: Standard method stable for all n, complex component in spectrum at n = 27. Predictor method unstable for n > 64.0 when gamma = 0.0. Predictor method stable for all n when gamma = 0.5. | | | |
| G | $\begin{bmatrix} 100 & 0 \\ 0 & 10 \end{bmatrix}$ | $\begin{bmatrix} 6 & -4 \\ 0 & 6 \end{bmatrix}$ | $\frac{-1}{1000}\begin{bmatrix} 60 & -40 \\ 900 & 600 \end{bmatrix}$ | -0.10, -0.56 |
| | Comments: Standard method stable for all n, complex component in spectrum at n = 27. Predictor method unstable for n > 64.0 when gamma = 0.0. Predictor method stable for all n when gamma = 0.9. | | | |
| H | $\begin{bmatrix} 15 & -5 \\ -5 & 105 \end{bmatrix}$ | $\begin{bmatrix} 6 & -4 \\ 0 & 6 \end{bmatrix}$ | $\frac{-1}{1550}\begin{bmatrix} 655 & -390 \\ 105 & 70 \end{bmatrix}$ | -0.10, -0.37 |
| | Comments: Standard method stable for all n, predictor method unstable for n > 81.0. Spectrum of both methods have complex components for finite range of n. | | | |
| J | $\begin{bmatrix} 15 & -5 \\ -5 & 105 \end{bmatrix}$ | $\begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix}$ | $\frac{-1}{1550}\begin{bmatrix} 655 & 30 \\ 105 & 90 \end{bmatrix}$ | -0.05, -0.43 |
| | Comments: Both standard and predictor methods stable for all n. Predictor method has complex components in spectrum for n > 25.0 | | | |
| K | $\begin{bmatrix} 15 & -5 \\ -5 & 105 \end{bmatrix}$ | $\begin{bmatrix} 6 & -4 \\ 0 & 6 \end{bmatrix}$ | $\frac{-1}{1550}\begin{bmatrix} 630 & -390 \\ 30 & 70 \end{bmatrix}$ | -0.06, -0.38 |
| | Comments: Both standard and predictor methods stable for all n. Predictor method has complex components in spectrum for n > 9.0. | | | |

**Figure 3.21** Comparison of capacitance, conductance and A matrices with eigenvalues.

Study E, where the G matrix is lower diagonal, results in the standard method being stable for all h, while the predictor method is still unstable for a finite range of h that lies between the value of the eigenvalues. This is an example where the standard method seems more desirable because of the stability properties. However, the accuracy plots show that for h where the predictor method is stable, the predictor method can also be much more accurate. When possible, Figure 3.14 indicates that the most desirable technique would be to use the predictor method where it is stable with gamma equal to zero. If this step size is too small for the desired speed of analysis, then the smallest gamma possible that insures stability would be needed for greatest accuracy.

The last five studies all have a similar diagonally dominant conductance matrix. The first two of these five, studies F and G, have a strictly diagonal capacitance matrix which result in each row of the A matrix being scaled, with no effect on the stability. The last two, studies J and K, have one element in the conductance matrix zeroed so that the conductance matrix becomes triangular. This seems to increase the range of h for which the methods are stable.

However, making an off-diagonal element zero should allow the matrices to behave more like diagonal matrices, and the first three studies, A, B, and C, have already demonstrated the behavior of a strictly diagonally G matrix.

In summary, both methods had either stability or accuracy problems if the conductance matrix was not diagonally dominant. In these studies, if the conductance matrix is diagonally dominant, then the standard Gauss-Seidel method is stable for all h, but a dominant conductance matrix is not sufficient to ensure that the predictor Gauss-Seidel method is stable for all h. In each case, however, if the method is stable for some $h_{\infty}$, then it is also stable for

all $h < h_{\infty}$. Therefore, a proper choice of gamma should be possible to make the predictor method as stable as the standard method but with increased accuracy for the same step size.

# CHAPTER 4

# PREMOS

## 4.1 Introduction

PREMOS (PREdiction Based simulator for MOS circuits) is an experimental circuit analysis program that is an attempt to bridge the gap between conventional circuit simulation and conventional logic simulation. PREMOS was developed by Wei at the University of Illinois and it evolved directly from MOTIS-C; but a number of basic changes were implemented in an attempt to increase the speed and accuracy. In addition to the predictor method described in the previous chapter, other changes include new data structures so that new analysis algorithms, such as analysis sequencing, could be implemented.

The subcircuit models built into the program are a major factor why PREMOS is faster than conventional circuit simulators such as SPICE2. However, if a circuit cannot be constructed from the built-in models, then PREMOS cannot be used for simulations. Thus, a user-specified subcircuit description, as well as an automatic partitioning algorithm, are needed to make the program more useful. Originally PREMOS was restricted to NMOS circuit simulations, but as part of the work reported in this thesis, several new subcircuits have been added to allow a limited CMOS circuit capability. These new subcircuits as well as how to use PREMOS are described in the following sections.

## 4.2 Input Circuit Description

The program PREMOS supplies the user with a variety of subcircuit models, henceforth referred to as primitives, that consist of a specific interconnection of transistors. The parameters (such as width to length ratio of the transistors and node capacitances) for each type of subcircuit are specified using the "model" card. The model card is also used to give a

name to each type of subcircuit, so that each primitive can be used repeatedly with a different set of parameters each time. The general format of the model card is

MODEL (modname) (primitive)(parameters)

After each necessary type of subcircuit has been described and named, the interconnection of the subcircuits is specified. In this section of the input the user gives each subcircuit a distinct name, specifies to which nodes the subcircuit is connected, and gives the model name specifying the proper type of subcircuit. The general format of the interconnect card is

(name) (nodes) (modname)

with one exception. This exception is when a subcircuit uses the SOURC primitive.

### *4.2.1 The SOURC Primitive*

The SOURC primitive is unique because it is used to describe voltage sources that switch between two levels, such as clock pulses, rather than an interconnection of transistors. The pulses are specified by a series of "1's" and "0's" following the model name on the interconnect card to indicate when the source is at the high or low voltage level, respectively. Linear interpolation is used to determine the voltage level during the rise and fall times when switching between levels. An example of this switching behavior is given in Figure 4.1. As long as $t_{low} + t_{rise}$ is the same as $t_{high} + t_{fall}$ (specified with the MODEL card), then the user should have no trouble as two consecutive digits will always have the same total duration. However, if the above condition does not hold, then care should be taken.

### *4.2.2 The CMOS Cells*

Four new CMOS primitives have been added to PREMOS. These are an inverter (Figure 4.2), a 2-input NAND (Figure 4.3), a 2-input NOR (Figure 4.4), and a variable number of transfer gates in series (Figure 4.5). The notation that follows is

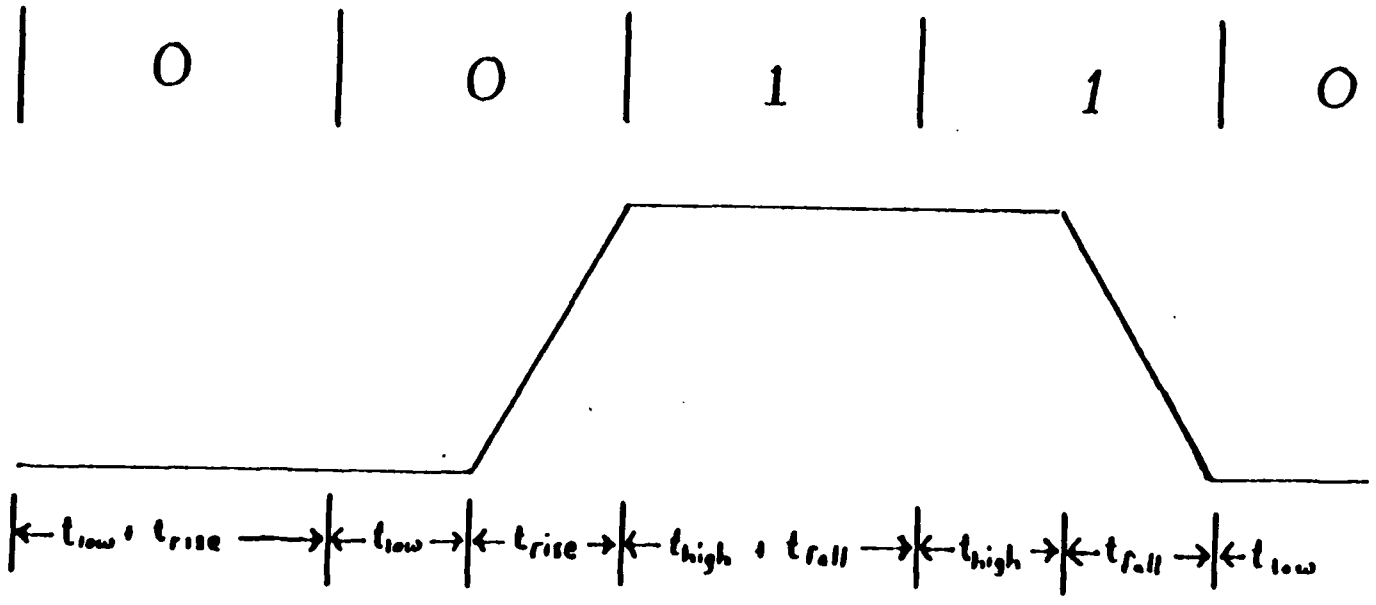



**Figure 4.1** **SOURC timing example.**

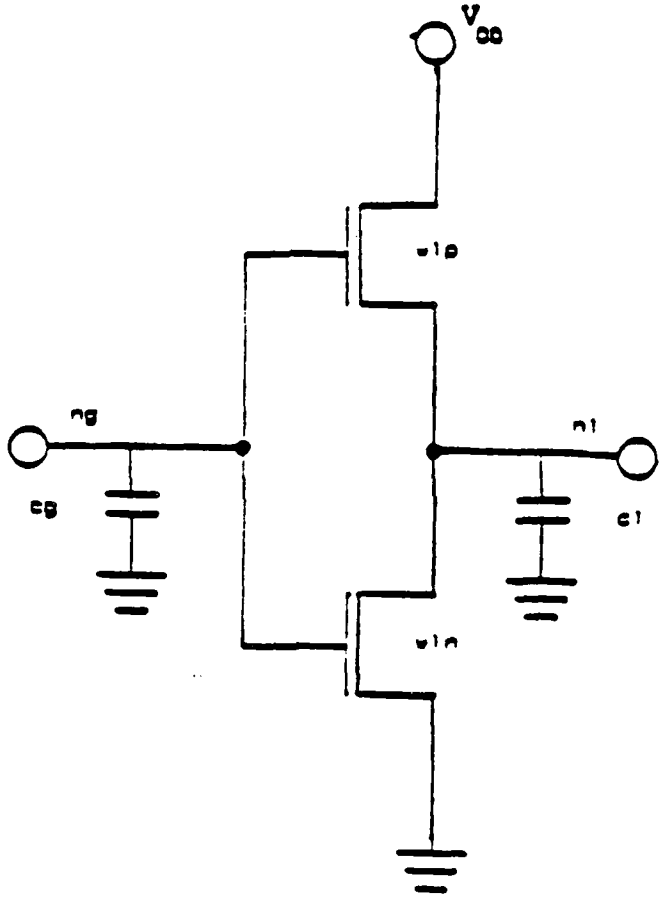



Figure 4.2 CMOS inverter.

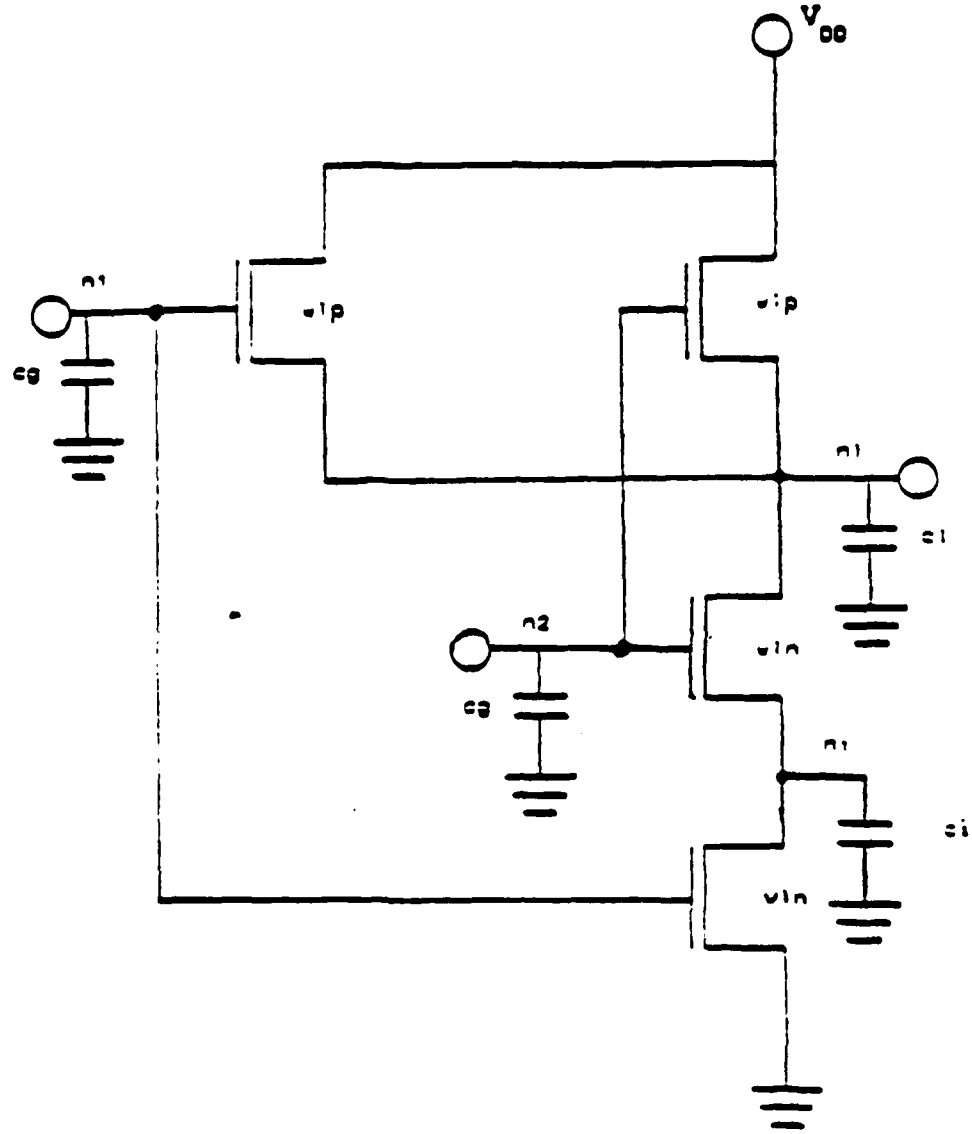



Figure 4.3 CMOS 2-input NAND gate.

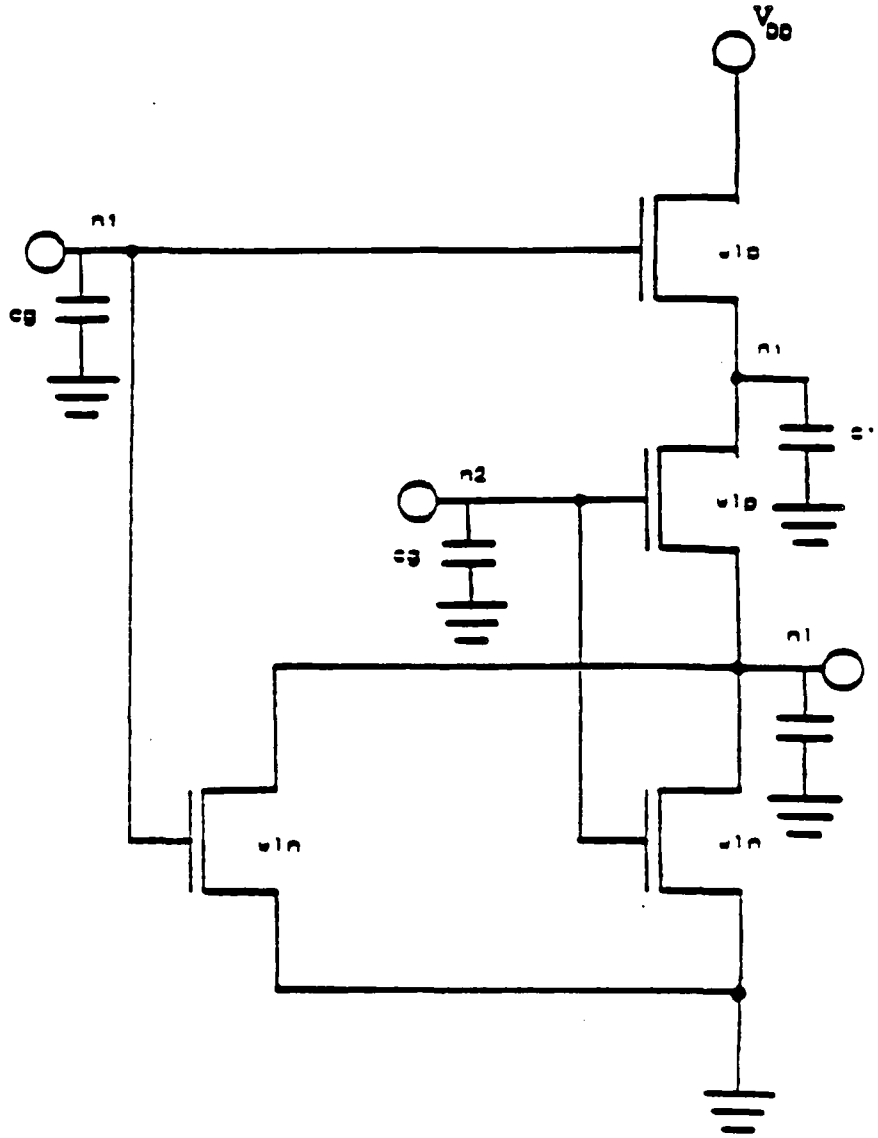



Figure 4.4 CMOS 2-input NOR gate.

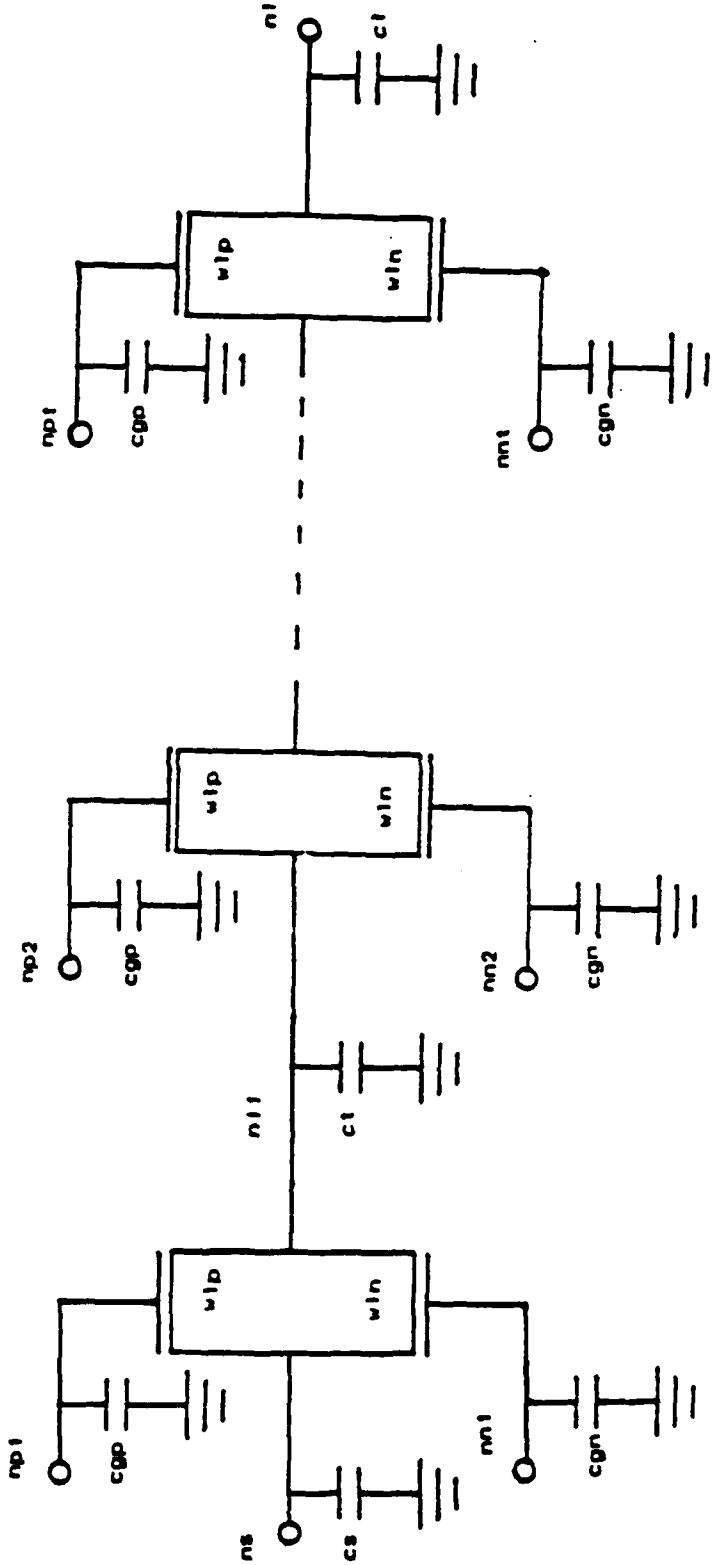



Figure 4.5 CMOS transfer gates.

wln = width-to-length ratio of the n-channel transistors

wlp = width-to-length ratio of the p-channel transistors

cg = gate (input) node capacitance

ci = internal node capacitance

cl = load (output) node capacitance

and, additionally, for the transfer gates

cs = source (input) node capacitance

cgn = gate of n-channel transistor capacitance

cgp = gate of p-channel transistor capacitance

ct = intertransfer gate transistor capacitance

nt = number of transfer gates

The parameters for the primitives are then specified in the following order:

```
CMINV  wln wlp cg cl
CMNOR  wln wlp cg ci cl
CMNAN  wln wlp cg ci cl
CMTFR  wln wlp cs cgn cgp ct cl nt
```

To specify circuit connections, the user must assign node numbers for each of these cells in the order:

```
CMINV  ng nl
CMNOR  n1 n2 ni nl
CMNAN  n1 n2 ni nl
CMTFR  ns nn1 np1 ni1 nn2 np2 . . . al
```

The first-time user should note that even internal node numbers for each cell must be specified even though internal nodes should never be connected to another cell.

To allow greater flexibility, the CMOS transfer gates have not been specifically linked to other type of logic as was done by Wei for NMOS transfer gates. However, large inaccuracies may result if the signal flow within the circuit is from the output node to the source node of a series of transfer gates instead of from the source to the output. Therefore, the user is encouraged to connect the output node of transfer gates only to the input node of other cells. It should also be noted that proper operation of CMOS transfer gates requires complementary control signals to the n-channel and p-channel transistors. Therefore, the use may need to add an inverter so that both polarities of the signal are available.

### 4.3 Control Commands

The control commands are input into PREMOS in the same file as the input data description. The nine basic commands are described in Appendix 3 of Reference [10]. The only changes have to do with the additional CMOS subcircuits and the modification to the predictor Gauss-Seidel method.

The OPT card is used to specify the number iterations allowed for each primitive for any single point in time. The number of dc iterations for the CMOS primitives can either be concatenated to the string given in Reference [10], or if a zero is input as the first number, then the CMOS dc iterations can be inputted directly. That is,

OPT itnan itnor ittrs itpul itlch itao itoa itmos itcin itcor itcan itctrs

or,

OPT 0 itcin itcor itcan itctrs

where itcin, itcor, itcan, and itctrs are the number of preset dc iteration for a CMOS inverter, a CMOS NOR gate, a CMOS NAND gate, and a CMOS transfer gate respectively and all other terms are as defined by Wei. The default number of iteration for any primitive not specified is one.

The CONTL card has simply added one more parameter, gamma, so that the modified predictor method can be used. If the predictor scheme is chosen then gamma may be specified. The default value of gamma is zero. The form of the CONTL card is now:

CONTL laten ltstp lpred gamma

with laten, ltstp, and lpred as defined by Wei.

### 4.4 Transistor Description

In addition to the input circuit description and control file already described, PREMOS requires a second data file describing the transistor characteristics. The modeling equations used are [15] and [16]:

$$I_{DS} = KP * \frac{1+\lambda V_{DS}}{1+\eta(V_{GS}-V_T)} * \left[(V_{GS} - V_T)\, V_{DS} - \frac{1}{2} V_{DS}^2\right] \tag{4.1}$$

for operation in the linear region and

$$I_{DS} = \frac{KP}{2} * \frac{1+\lambda V_{DS}}{1+\eta(V_{GS}-V_T)} * (V_{GS} - V_T)^2 \tag{4.2}$$

for operation in the saturation region with

$$V_T = V_{T0} + \Delta V_T \tag{4.3}$$

and

KP = intrinsic transconductance

$\lambda$ = channel length modulation parameter

$\eta$ = mobility reduction parameter

$V_{T0}$ = threshold voltage at dc bias

$\Delta V_T$ = threshold voltage change due to substrate bias voltage change

These parameters for each type of transistor must be in a file named "datai.dat." Each transistor has a set of 24 parameters, the first four of which are KP, $\lambda$, $\eta$, and $V_{T0}$. The next 20 are $\Delta V_T$ as represented in tabular form as a function of the source to substrate voltage. The first set of 24 is used to describe all of the NMOS enhancement mode transistors. The second set of 24 is used to describe the NMOS depletion mode transistors and should have a negative $V_{T0}$. The third group of 24 is for the PMOS enhancement mode transistor and should also have a negative $V_{T0}$. Finally, the last group of 24 is reserved for PMOS depletion mode transistors, even though no primitives have been implemented yet that use this type of transistor.

### 4.5 Output Processing

In addition to being able to generate plots on standard printer, PREMOS is also capable of generating output suitable for input to a graphics plotter. Currently, the SEND command will generate an output data file named "plfile.dat" which can then be used by a program named "graf" for making graphs on Tektronics 4010 series terminals. All the user has to do is execute graf, which will read in the file plfile.dat and then prompt the user for all necessary additional input.

### 4.6 Interactive Session Commands

At this point it is appropriate to demonstrate the use of PREMOS to analyze a particular circuit. The circuit chosen for this is a ring oscillator constructed from CMOS gates as shown in Figure 4.6. A file named "cring.dat" describing the circuit and giving the control commands for this circuit is given in Figure 4.7. The transistor data file (always named "datai.dat") is given if Figure 4.8. At the time of this writing, PREMOS has been developed in conjunction with the BSD 4.1 version of the UNIX™ operating system. Given that the executable code for PREMOS is contained in the file named "prec.ext" and letting bold face

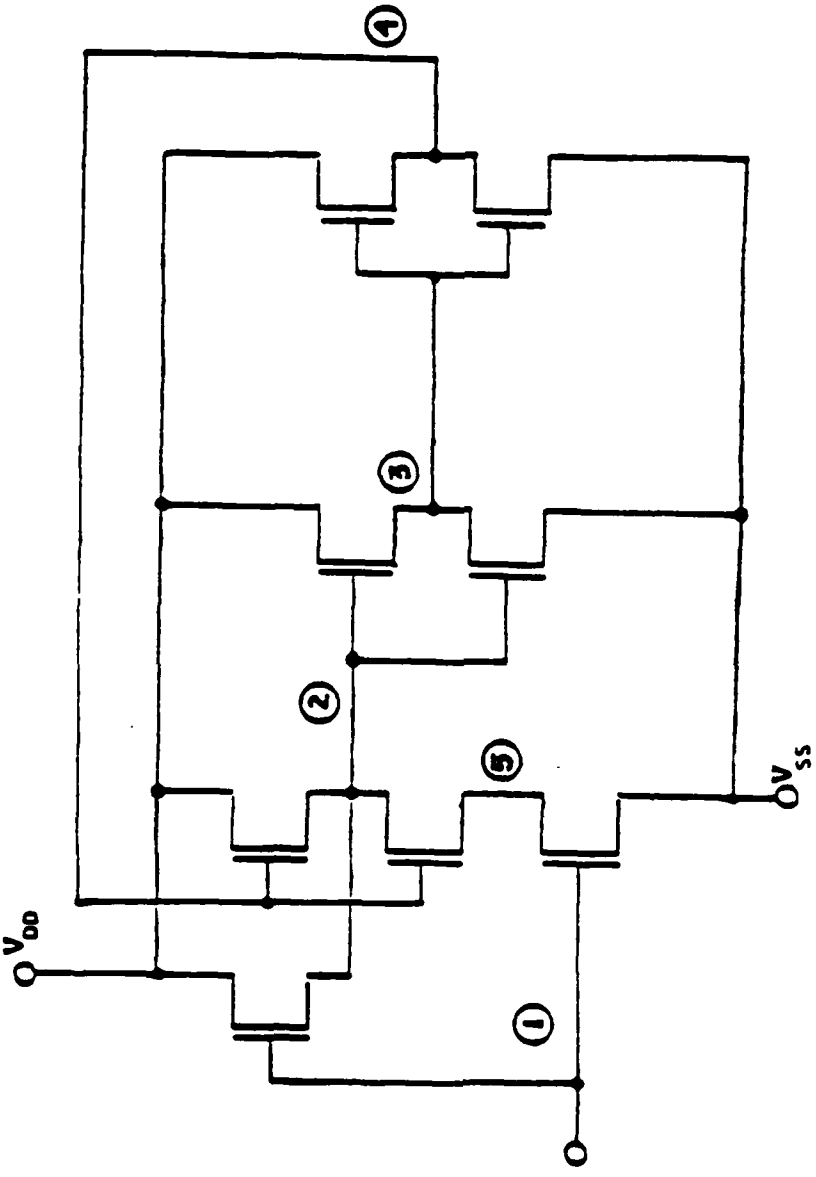



Figure 4.6 CMOS ring oscillator.

```
3 stage cmos ring oscillator
model inv cminv(2 2 20f 20f)
model nand cmnan(2 1 20f 15f 20f)
model clk soure(5 0 10n 5n 10n 5n)
x1 1 4 5 2 nand
x2 2 3 inv
x3 3 4 inv
c1 1 0 clk 0 1 0 0 1 0
opt 0 3 3 3 3
dc
cntl 1 1 1
time 40n in
plot 1 2 3
send 1 2 3
v+ 5
end
*
```

Figure 4.7 Input circuit and control file for CMOS oscillator.

```
2.000d-05 1.000d-03 1.000d-03 1.000d+00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
2.000d-05 1.000d-03 1.000d-03 -2.000d00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
2.000d-05 1.000d-03 1.000d-03 -1.00d+00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
2.000d-05 1.000d-03 1.000d-03 2.000d+00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00 0.000d+00
```

Figure 4.8 Transistor process parameter file for ring oscillator.

denote what the user needs to type in response to various prompts, then a typical analysis of a circuit would proceed as follows:

```
% prec.ext <return>
 circuit file name?  cring.dat <return>
 listing file name?  cring.out <return>
%
```

At this time, the output data is in the file "cring.out" and can be listed on a terminal or printed on paper. PREMOS has also created a new file with the name "plfile.dat" for use with the graphing routines. If "plfile.dat" already existed, it is overwritten with the new output.

The plotting routines have been written using PLOT10 software for use with Tektronics 4010 series terminals. Assuming the user is now logged on to an appropriate graphics terminal, the session could proceed as follows:

```
% graf
 want hard copy? yes = 1, no = 0
 1 <return>
 1st symbol type=? 0(no),1(0),2(x),5(°),8(+),<11
 1st line type=? 1-4(dash),0(solid),-1(no),<11
 sequential line type: 0 1 2 3 4 23 34, <11
 0 0 <return>
 2nd symbol type=?  2nd line type=?
 0 1 <return>
          .
          .
          .
          .
          .
          .
          .
          .
```

After the user has selected the desired style of line for each of up to seven output voltages (previously selected with the SEND command in the input file), the waveforms will be plotted. When the user is ready to continue with other commands, the "return" key must be pressed to obtain another prompt from the computer. As can be seen above, a number of possible plotting styles are given to the user in the prompt for the first waveform. For other

possibilities, refer to the PLOT10 manuals. The graphics output resulting from the input file "cring.dat" is shown in Figure 4.9. Other sample circuits are include in the appendix.

### 4.7 Discussion

PREMOS is a fairly easy program to use provided the circuit can be described with the primitives provided. Furthermore, after learning how to add one primitive, it is fairly easy to add new primitives as the need arises. However, in the present implementation adding new primitives increases the size of the program, while for speed it is desirable to keep the program small. Since the primitives are used mainly in identifying the topology of a circuit, it may be desirable to change the structure of the program so that the topology of the circuit is stored in an intermediate data file. While allowing a much larger selection of primitives to be maintained, this would also have the benefit that different control commands inputs could be used without the overhead of always analyzing the basic structure of the circuit. This could be valuable if the predictor method was ever found to be unstable for a particular circuit.

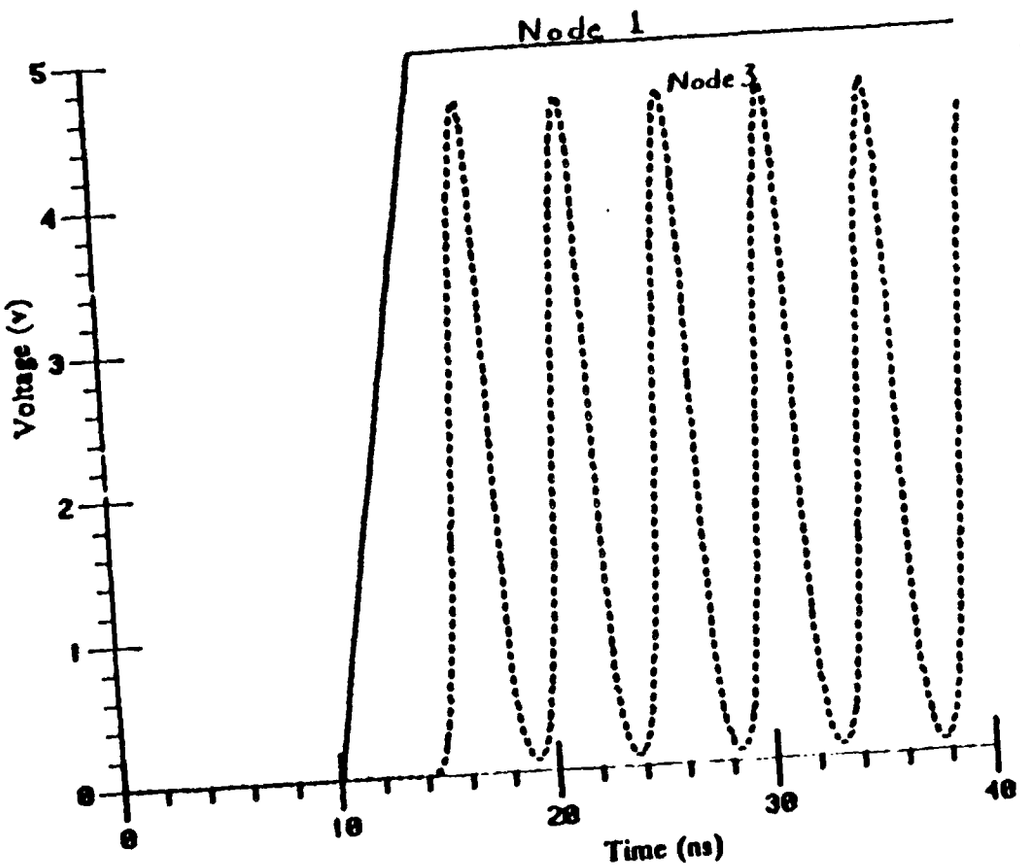



**Figure 4.9** Node voltage plots for CMOS ring oscillator.

# CHAPTER 5

## Conclusions

Performing circuit level simulation at logic level speeds is an active area of research. The methods used in the program PREMOS to increase the speed of analysis include analysis sequencing, latency checking, and circuit partitioning. While these techniques do decrease the amount of computer time required, some important restrictions must be placed on the type of circuits allowed. As implemented in PREMOS, one major requirement is that a circuit must be composed of available unidirectional subcircuits. The unidirectional requirement means that feedback within a circuit requires special numerical algorithms. This work has attempted to show that the predictor Gauss-Seidel method is a viable approach.

While the predictor Gauss-Seidel method is not in general stable for all positive time step sizes, it has been shown to be stable for some interval of step sizes in each of seven studies. Perhaps just as important is the comparison between the predictor and the (commonly used) standard Gauss-Seidel method. With the proper choice of a parameter, the predictor method can be made stable over as wide a range of step size as the standard method. The major drawback with this is that there is no way to determine the optimum value of this parameter at present.

PREMOS is a program that is accurate enough for many applications. However, for future experimental work it may be helpful to break the program into smaller, interacting functional blocks. At the present time PREMOS contains over 4500 lines of code. While this is not nearly as large a program as SPICE, continued growth could certainly slow the speed of analysis. Breaking the program into smaller functional blocks would have the disadvantage that more information would have to be stored for the blocks to be able to

communicate, but would allow the user to change one part of his input data without having to necessarily execute the whole program again. Further experimental work would also be simplified if parts of the program could be more easily modified without affecting unrelated areas of analysis.

# APPENDIX

## Circuit Examples

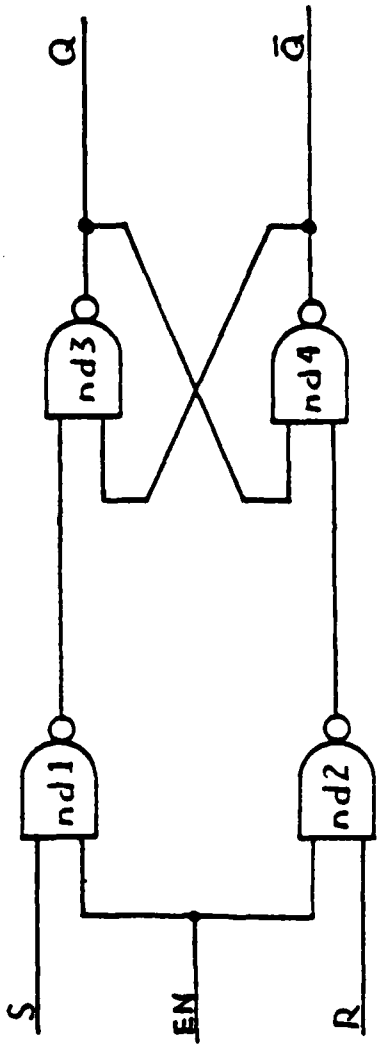



**Figure A.1 Gated R-S Flip-Flop.**

```
cmos RS flip-flop
model nand cmnan(2 1 15f 10f 50f)
model clk sourc(5 0 10n 5n 10n 5n)
*
nd1   1   3   8   4 nand
nd2   2   3   9   5 nand
nd3   4   7 10   6 nand
nd4   5   6 11   7 nand
*
ss 1 0 clk 1 1 0 0 1 1 0 0 0 0 1
rr 2 0 clk 0 0 1 1 1 1 0 0 1 0 1
en 3 0 clk 1 1 1 1 1 1 1 0 1 0 1
*
opt 0 3 3 3 3
dc
cont1 1 1 1
time 120n 1n
send 1 2 6 7
v+ 5
end
*
```

**Figure A.2** Input Data File for Gated R-S Flip-Flop.

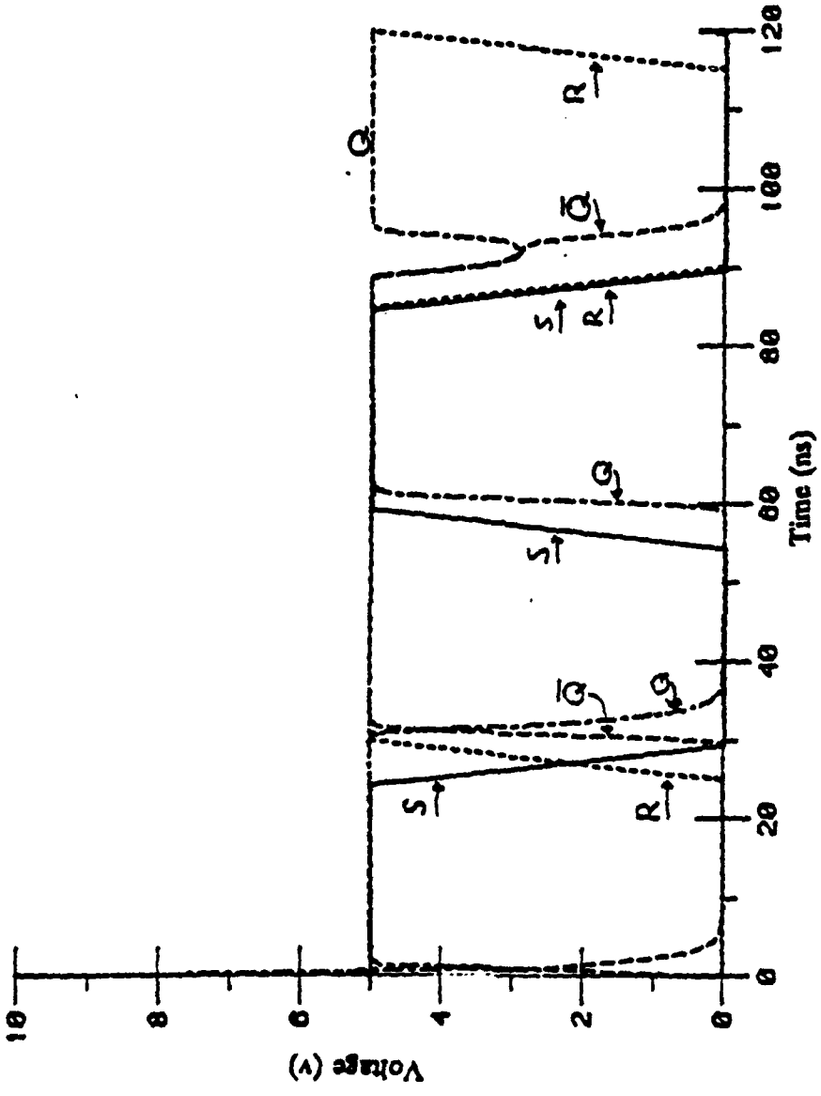



Figure A.3 Voltage Plots for Gated R-S Flip-Flop.

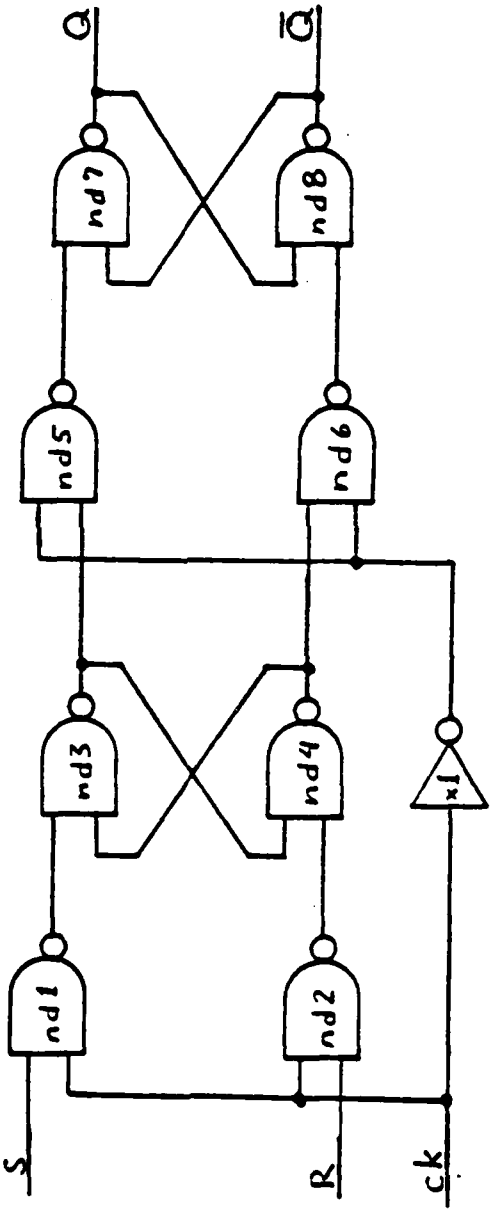



Figure A.4 Master-Slave R-S Flip-Flop.

```
cmos master-slave RS flip-flop
model inv cminv(2 2 10f 20f)
model nand cmnan(2 1 15f 10f 50f)
model clk sourc(5 0 10n 5n 10n 5n)
*
x1 3 8 inv
nd1  1  3 13  4 nand
nd2  2  3 14  5 nand
nd3  4  7 15  6 nand
nd4  5  6 16  7 nand
nd5  6  8 17  9 nand
nd6  7  8 18 10 nand
nd7  9 12 19 11 nand
nd8 10 11 20 12 nand
*
ss 1 0 clk 0 0 1 1 0 0 0 0 0 1 0 1 0
rr 2 0 clk 0 0 0 0 1 1 0 0 0 0 1 0 1
ck 3 0 clk 1 0 1 0 1 0 1 0 1 0 1 0 1
*
opt 0 3 3 3 3
dc
cont1 1 1 1
time 120n 1n
send 1 2 3 11 12
v+ 5
end
*
```

**Figure A.5** Input Data File for Master-Slave R-S Flip-Flop.

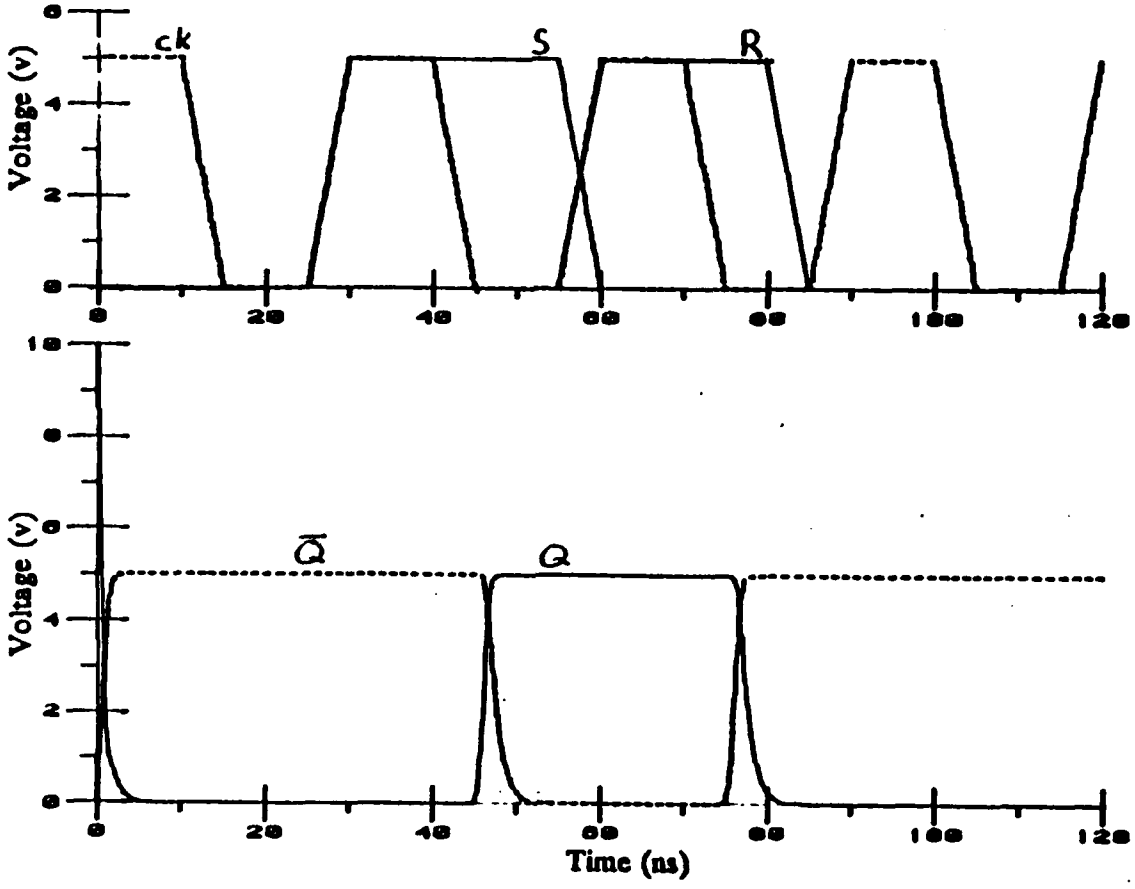



**Figure A.6** Voltage Plots for Master-Slave R-S Flip-Flop.

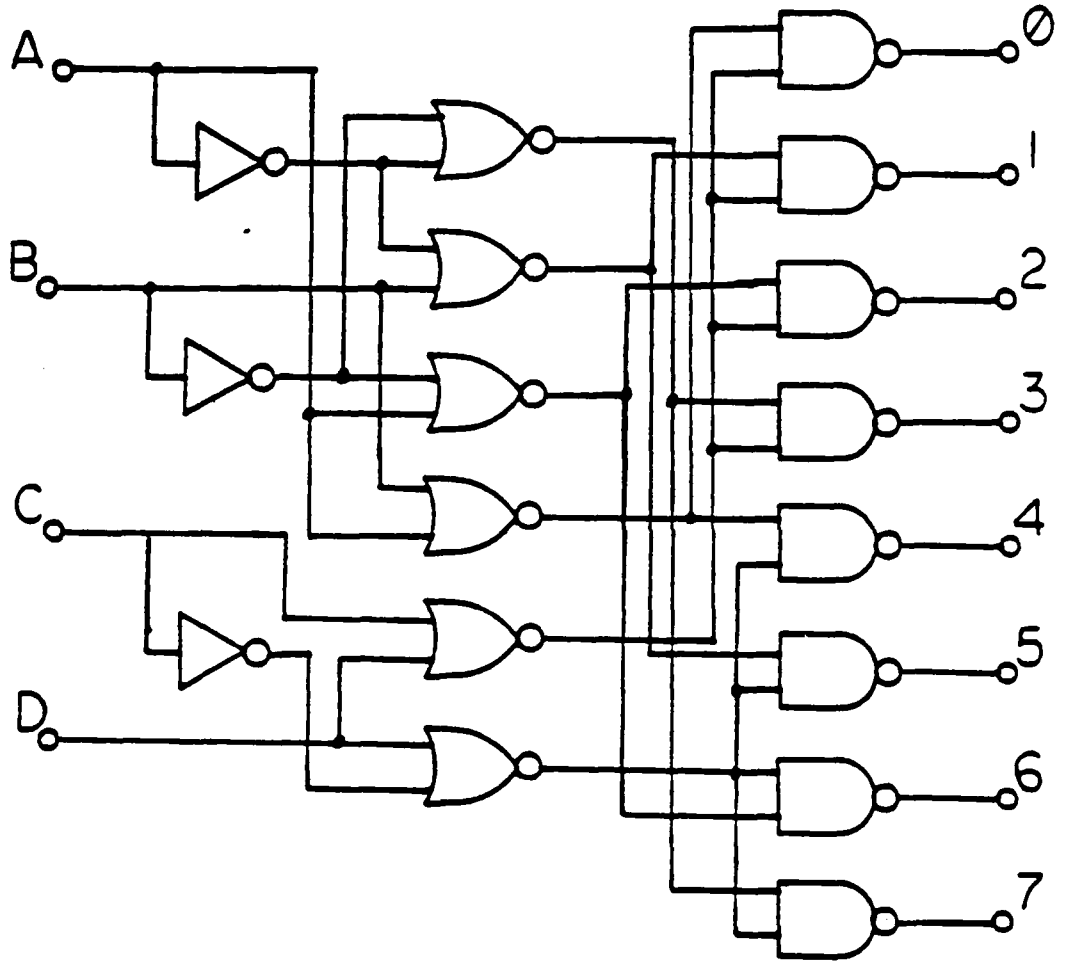



Figure A.7 Binary-to-Octal Decoder.

```
cmos binary-to-octal decoder
model inv cminv(2 2 10f 20f)
model nand cmnan(2 1 15f 10f 50f)
model nor cmnor(1 2 15f 10f 20f)
model clk sourc(5 0 10n 5n 10n 5n)
*
x1 1 5 inv
x2 2 6 inv
x3 3 7 inv
*
nr1 5 6 22 8 nor
nr2 5 2 23 9 nor
nr3 6 1 24 10 nor
nr4 2 1 25 11 nor
nr5 3 4 26 12 nor
nr6 7 4 27 13 nor
*
nd1 12 11 28 14 nand
nd2  9 12 29 15 nand
nd3 12 10 30 16 nand
nd4  8 12 31 17 nand
nd5 11 13 32 18 nand
nd6  9 13 33 19 nand
nd7 10 13 34 20 nand
nd8  8 13 35 21 nand
*
ck1 1 0 clk 0 1 0 1 0 1 0 1 0 1 0 1
ck2 2 0 clk 0 0 1 1 0 0 1 1 0 0 1 1
ck3 3 0 clk 0 0 0 0 1 1 0 0 0 1 1 1
ck4 4 0 clk 0 0 0 0 0 0 1 1 1 1 0 0
*
opt 0 3 3 3 3
dc
cont1 1 1 1
time 60n 1n
send 14 15 16
v+ 5
end
*
```

Figure A.8 Input Data File for Binary-to-Octal Decoder.

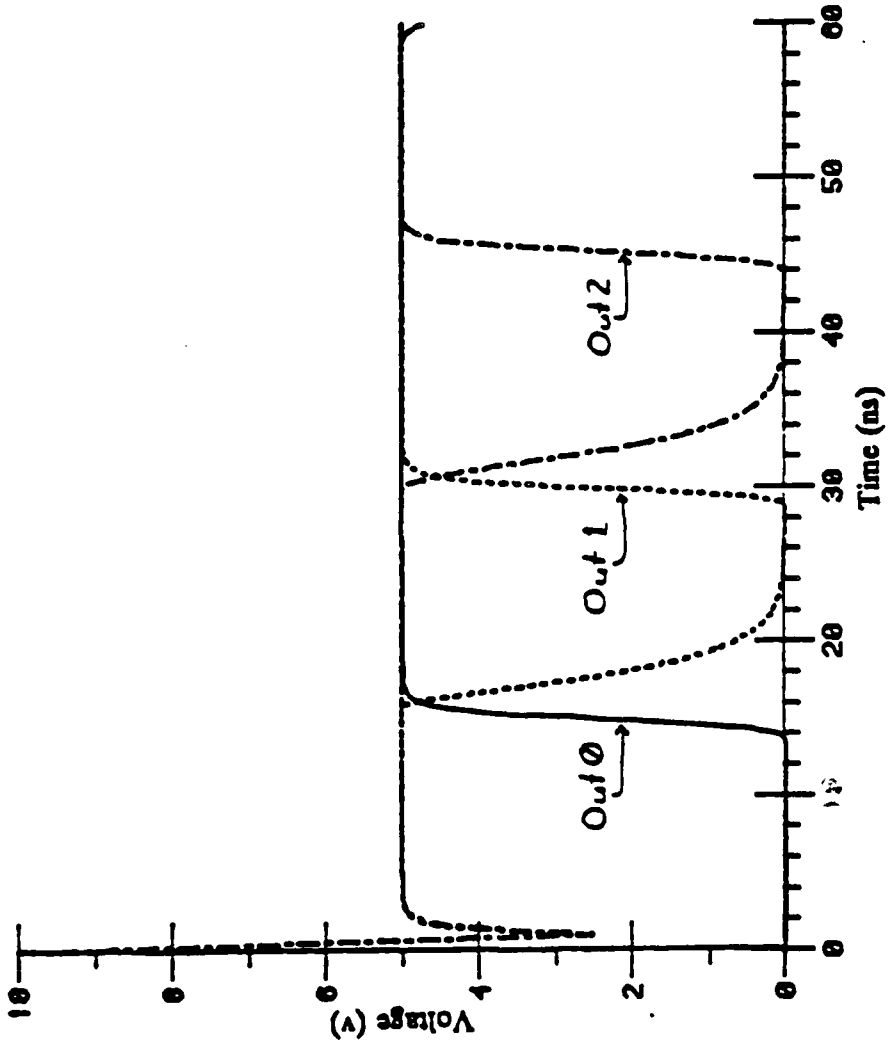



Figure A.9 Voltage Plots for Binary-to-Octal Decoder.

# REFERENCES

[1] L. W. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits," ERL Memo., No. ERL-M520, University of California, Berkeley, May, 1975.

[2] B. R. Chawla, H. K. Gummel and P. Kozak, "MOTIS - An MOS Timing Simulator," IEEE Trans. on Circuits and Systems, Vol. CAS-22, No. 12, pp. 901-910, December, 1975.

[3] S. P. Fan, M. Y. Hsueh, A. R. Newton and D. O. Pederson, "MOTIS-C: A New Circuit Simulator for MOS LSI Circuits," IEEE Int. Symp. on Circuits and Systems, Arizona, pp. 700-703, April, 1977.

[4] N. Tanabe, H. Nakamura and K. Kawkits, "MOSTAP: An MOS Circuit Simulator for LSI Circuits," Proc. IEEE Int. Symp. on Circuits and Systems, pp. 1035-1038, April, 1980.

[5] I. N. Hajj, "Sparsity Considerations in Network Solution by Tearing," IEEE Trans. on Circuits and Systems, Vol. CAS-27, No. 5, pp. 357-366, May, 1980.

[6] P. Yang, "An Investigation of Ordering, Tearing, and Latency Algorithms for the Time Domain Simulation of Large Circuits," CSL Report R-891, University of Illinois at Urbana-Champaign, August, 1980.

[7] E. Lelarasmee, A. E. Ruehli, and A. L. Sangiovanni-Vincentelle, "The Waveform Relaxation Method for Time Domain Analysis of Large Scale Integrated Circuits," IEEE Trans. Computer Aided Design, Vol. CAD-1, No. 3, pp. 131-145, July, 1982.

[8] A. R. Newton, "The Simulation of Large Scale Integrated Circuits," ERL Memo. No. ERL-M78/52, University of California, Berkeley, July, 1978.

[9] Giovanni De Micheli and Aberto Sangiovanni-Vincentelle, "Characterization of Integration Algorithms for the Timing Analysis of MOS VLSI Circuits," Int. J. Circuit Theory Appl., Vol. 10, pp. 299-309, 1982.

[10] You-Pang Wei, "Large-Scale Circuit Simulation," CSL-Report R-977, University of Illinois at Urbana-Champaign, December, 1982.

[11] C. A. Desoer, Notes for a Second Course on Linear Systems, Princeton, New Jersey: D. Van Nostrand, 1970.

[12] L. O. Chua and P. M. Lin, Computer Aided Analysis of Electronic Circuits, Englewood Cliffs, New Jersey: Prentice-Hall, 1975.

[13] S. M. Selby, CRC Standard Mathematical Tables, Cleveland, Ohio: CRC Press, 1975.

[14] G. Strang, Linear Algebra and Its Applications, New York, New York: Academic Press, 1980.

[15] D. A. Hodges and H. G. Jackson, Analysis and Design of Digital Integrated Circuits, New York, New York: McGraw Hill, 1983.

[16] D. R. Alexander, et.al., "SPICE2 MOS Modeling Handbook," Report BDM/A-77-071-TR, BDM Corporation, Albuquerque, New Mexico, 1977.